



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ

РОБОТОТЕХНИКА И КОМПЛЕКСНАЯ АВТОМАТИЗАЦИЯ (РК)

КАФЕДРА

РК6 «СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ»

Отчет по домашнему заданию

Применение метода последовательности ближайших соседей для улучшения алгоритма пчелиной колонии

Студент

**Абидоков Р. Ш.
РК6-21М**

Преподаватель

Карпенко А. П.

2022 г.

Постановка задачи

Пусть $X = \{x_1, x_2, \dots, x_D\} \in R^D$ – вектор действительных чисел размерностью $D, D \geq 1$; $f(X): R^D \rightarrow R$ – скалярная целевая функция векторного аргумента X . Задачей многомерной безусловной оптимизации называется задача поиска вектора X^* , удовлетворяющего уравнению

$$\min_{X \in R^D} f(X) = f(X^*) = f^*. \quad (1)$$

Описание алгоритма

В данной работе для численного поиска X^* , удовлетворяющего (1), используется модифицированный алгоритм пчелиной колонии с использованием последовательности ближайших соседей (далее MNNABC), предложенный в [2]. Основными отличиями от оригинального алгоритма пчелиной колонии (далее ABC), предложенного в [1], служат:

- попеременное использование на стадии *employed bee* двух поисковых стратегий, основанных на модифицированном методе ближайших соседей (MNN);
- замена метода "рулетки" на стадии *onlooker bee* для выбора особей, подлежащих дальнейшему уточнению, на оригинальный, также основанный на MNN.

Таким образом, на обеих основных стадиях алгоритма необходимо построение последовательности $MS_i = \{X_i^0, X_i^1, \dots, X_i^{m_i}\}$ ближайших соседей для особи X_i , принадлежащей популяции $P(t) = \{X_1, X_2, \dots, X_{SN}\}$ мощностью SN . Приведем алгоритм построения такой последовательности, предложенный в [2]:

1. В качестве нулевого элемента последовательности X_i^0 берется исходная особь X_i .
2. Из всех особей популяции $P(t) = \{X_1, X_2, \dots, X_{SN}\}$ выбираются множество особей $S_i = \{X | X \in P(t) \wedge f(X) < f(X_i)\}$, т.е. все особи, имеющие целевую функцию меньшую, нежели X_i .
3. В последовательность MS_i добавляется особь из множества S_i , ближайшая к X_i в смысле евклидова расстояния.
4. Пункты 2 и 3 циклически повторяются для последнего добавленного в последовательность элемента до тех пор, пока множество S_i не пусто. После этого последовательность $MS_i = \{X_i^0, X_i^1, \dots, X_i^{m_i}\}$ с количеством элементов $m_i + 1$ считается сформированной.

Иллюстрация данного алгоритма приведена на Рис. 1 (взято из оригинальной статьи). Очевидно, что последний элемент последовательности один и тот же для всех особей популяции и является лучшим во всей популяции. Если же особь X_i является лучшей в популяции, $m_i = 0$ и её последовательность ближайших соседей состоит только из самой особи.

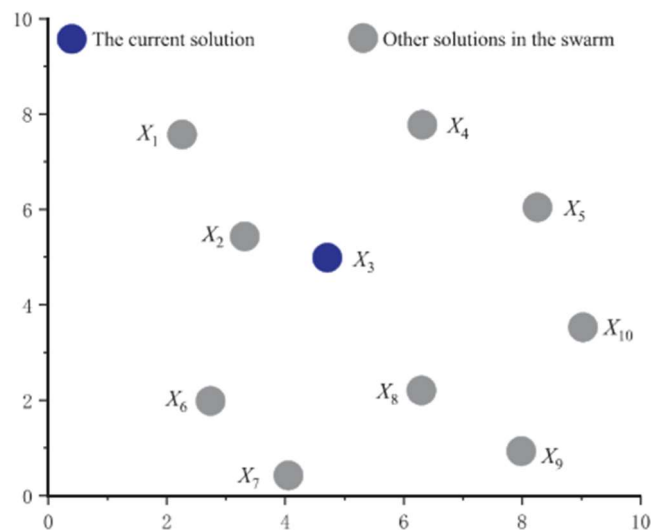


Рис. 1.1.

Выбор особи популяции X_i для построения множества MS_i

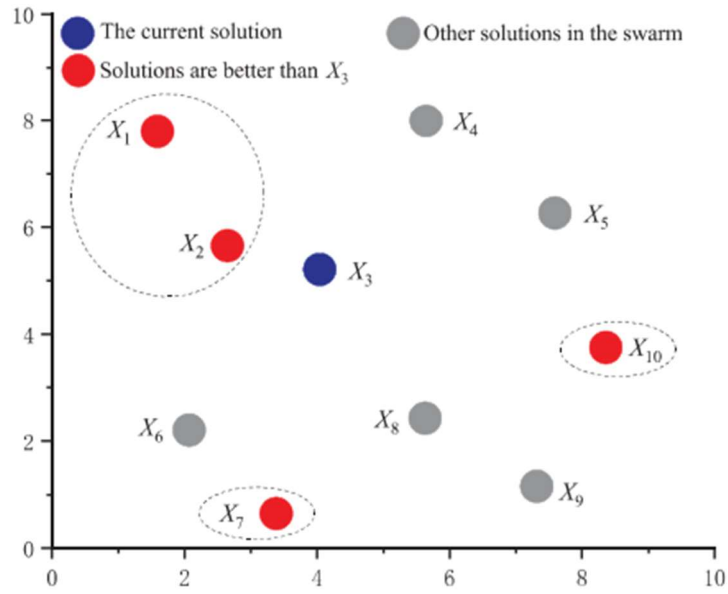


Рис. 1.2.
Построение множества S_i

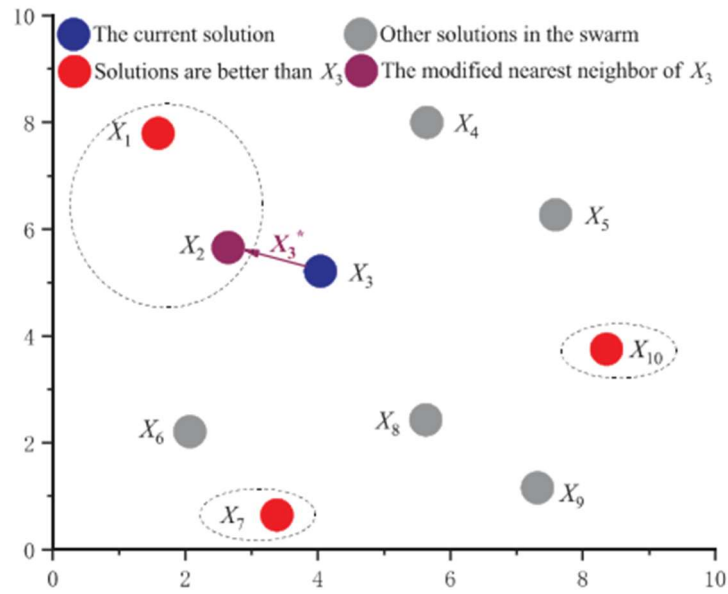


Рис. 1.3.
Выбор из множества S_i ближайшего соседа

Как было сказано выше, последовательности MS_i используются в предложенных авторами стратегиях поиска, используемых для получения особи V_i на следующей итерации алгоритма. Опишем первую из этих стратегий. Компоненты векторов $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ находятся как

$$v_{i,j} = mcx_{i,j} + \phi_{i,j}(x_{best,j} - x_{k,j}), \quad (2)$$

$$mcx_{i,j} = \frac{\sum_{h=0}^{m_i} x_{i,j}^h}{m_i + 1},$$

где $i = 1, 2, \dots, SN$, j – случайное целое число в интервале $[1, D]$, $\phi_{i,j}$ – случайное число в интервале $[-1, 1]$, $x_{best,j}$ – j -я компонента лучшей особи популяции X_{best} , $x_{k,j}$ – j -я компонента вектора X_k , случайно выбранного из всей популяции,

$MCX_i = (mcx_{i,1}, mcx_{i,2}, \dots, mcx_{i,D})$ – центр (вектор средних значений) последовательности MS_i , $mcx_{i,j}$ – j -я компонента MCX_i , $x_{i,j}^h$ – j -я компонента вектора X_i^h , принадлежащего последовательности MS_i .

В свою очередь, вторая стратегия описывается формулой

$$v_{i,j} = \begin{cases} x_{best,j} + \phi_{i,j}(x_{i,j}^{h+1} - x_{i,j}^h), & \text{если } m_i \geq 1 \\ x_{best,j} + \phi_{i,j}(x_{best,j} - x_{k,j}), & \text{если } m_i = 0, \end{cases} \quad (3)$$

где $x_{i,j}^h, x_{i,j}^{h+1}$ – j -е компоненты соответственно особей X_i^h, X_i^{h+1} , случайно выбранных из последовательности MS_i .

Вместе эти две стратегии образуют пул стратегий $SP = \{(2), (3)\}$, из которых выбирается стратегия для следующего обновления особи X_i по принципу

$$SX_i = \begin{cases} SX_i, & \text{если } f(V_i) > f(X_i) \\ SP - \{SX_i\}, & \text{если } f(V_i) \leq f(X_i) \end{cases} \quad (4)$$

Формула инициализация начальных положений особей X_1, X_2, \dots, X_{SN} популяции:

$$x_{i,j} = x_{min,j} + rand(0, 1) * (x_{max,j} - x_{min,j}) \quad (5)$$

Псевдокод алгоритма MNNABC приведен в Алгоритмах 1, 2, 3.

Алгоритм 1. Стадия employed bee

```

for each  $X_i$  do
  Сгенерировать  $V_i$  по соответствующей стратегии  $SX_i$ ;
  Вычислить  $f(V_i)$ ;
  if  $f(V_i) < f(X_i)$  then
     $X_i = V_i$ ;
     $trial_i = 0$ ;
  else
     $trial_i = trial_i + 1$ ;
  end
  Обновить стратегию  $SX_i$  в соответствии с (4);
end

```

Алгоритм 2. Стадия onlooker bee

```

for  $i = 1$  to  $SN$  do
  Получить последовательность  $MS_i$  для  $X_i$ ;
  Случайно выбрать  $X_i^{ri}$  из  $MS_i$ ;
  Найти индекс  $h$  в популяции для  $X_i^{ri}$  ( $X_h == X_i^{ri}$ );
  Сгенерировать  $V_h$  по соответствующей стратегии  $SX_h$ ;
  Вычислить  $f(V_h)$ ;
  if  $f(V_h) < f(X_h)$  then
     $X_h = V_h$ ;
     $trial_h = 0$ ;
  else
     $trial_h = trial_h + 1$ ;
  end
  Обновить стратегию  $SX_h$  в соответствии с (4);
end

```

Алгоритм 3. MNNABC

```

Инициализировать популяцию мощностью  $SN$  в соответствии с (5);
 $X_{best}^{prev} = X_{best}$ ;
 $n = 0$ ;
while  $n < n_{lim}$  do
  Выполнить Алгоритм 1 (employed bee);
  Выполнить Алгоритм 2 (onlooker bee);
  if  $\max\{trial_i\} > limit$  then
    Заменить  $X_i$  новым в соответствии с (5) (scout bee);

```

```

triali = 0;
end
if abs( $X_{\text{best}} - X_{\text{best}}^{\text{prev}}$ ) <  $\delta_f$  then
    n = n + 1;
else
    n = 0;
end
 $X_{\text{best}}^{\text{prev}} = X_{\text{best}}$ ;
end

```

Программная реализация

Предложенный алгоритм был реализован на языке программирования Python 3.8 с использованием библиотеки numpy.

Тестирование проводилось на сферической функции размерности $D = 2$ с центром в точке $(0, 0)$, критерием останова служила стагнация решения более 20-и итераций (т.е. $n_{\text{lim}} = 20$) при $\delta_f = 10^{-6}$. Положения особей популяции на различных итерациях алгоритма приведены на Рис. 2.

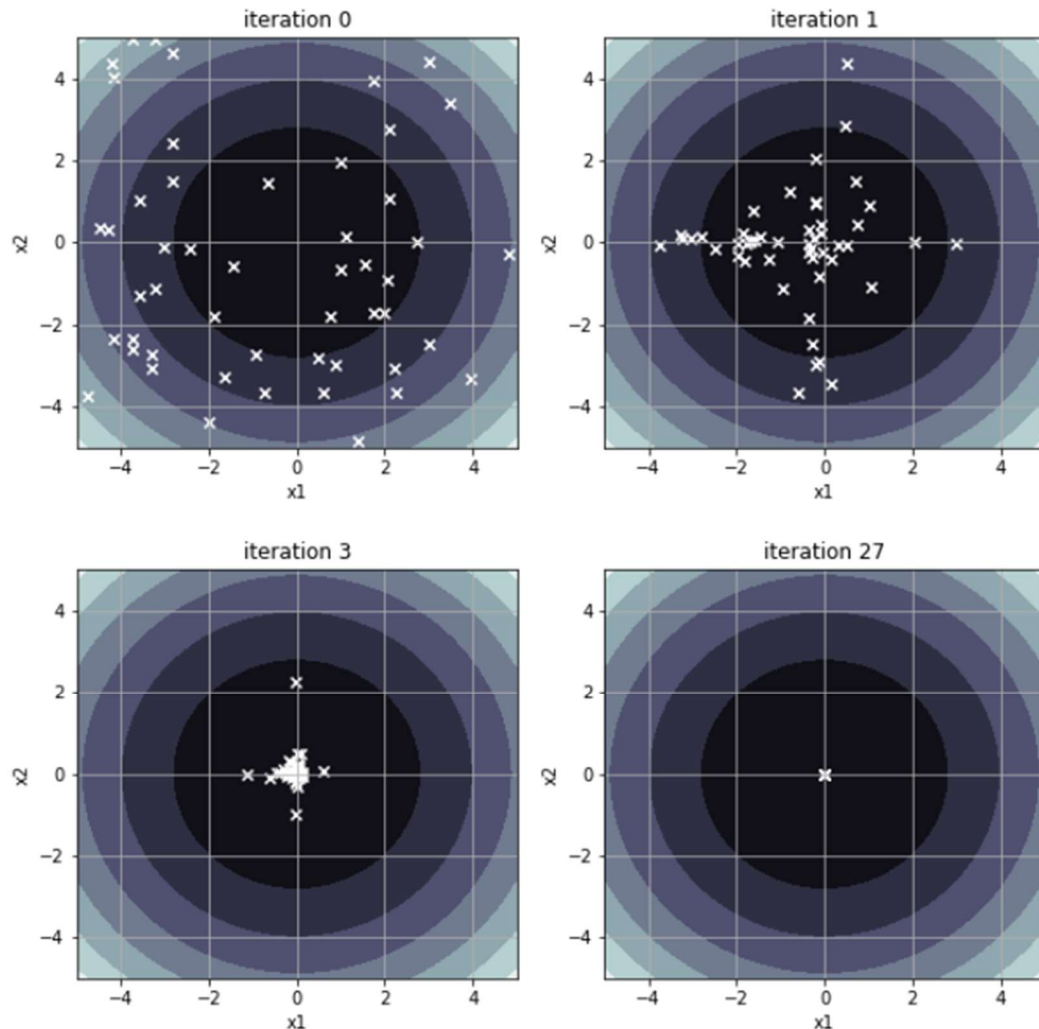


Рис. 2.

Положения особей X_i на различных итерациях алгоритма

Найден минимум $X^* = (-1.988 * 10^{-10}, 6.944 * 10^{-1})$ со значением целевой функции $f^* = 5.217 * 10^{-1}$. Соответственные погрешности $\delta X^* = 7.223 * 10^{-1}$, $\delta f^* = 5.217 * 10^{-1}$.

Вычислительный эксперимент

Вычислительный эксперимент проводился для двух функций – Растригина, заданной уравнением

$$f(X) = \sum_{j=1}^{|X|} (x_j^2 - 10 \cos(2\pi x_j) + 10) + f_0, \quad (6)$$

и Розенброка, заданной уравнением

$$f(X) = \sum_{j=1}^{|X|-1} \left(100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2 \right) + f_0, \quad (7)$$

для различных значений размерности $|X| = 2, 4, 8, 16, 32$.

При этом начальные распределения особей популяции задавались так, что $x_j \in [-\alpha, \alpha]$ для функции Растригина, и $x_j \in [1 - \alpha, 1 + \alpha]$ для функции Розенброка (поскольку её глобальный минимум расположен в точке $(1.0)^{|X|}$). Значения величины α равны 5, 1, 0.2 для функции Растригина и 6, 2, 0.2 для функции Розенброка соответственно. Количество особей в популяции $SN = 50$.

Критериями останова алгоритма служили стагнация вычислительного процесса на протяжении 20 итераций при $|\delta f| \leq 10^{-6}$.

Для каждой размерности $|X|$ проводились мултистарты в количестве 100. Метриками качества работы алгоритма служили абсолютная погрешность по лучшему значению целевой функции δf^* , нормы абсолютная погрешность по компонентам вектора лучшего решения δX^* , а также средние значения абсолютных погрешностей по целевой функции $\overline{\delta f^*}$ и норме вектора $\overline{\delta X^*}$, а также вероятность локализации глобального минимума p_{glob} . Критерием локализации глобального минимума служит $\delta f^* \leq 10^{-2}$.

Результаты вычислений приведены в Табл. 1-6.

Таблица 1. Функция Растригина, $\alpha = 5$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	3.10e-10	1.88e-08	0.00e+00	1.80e-14
4	1.00	8.77e-09	1.09e-07	0.00e+00	4.02e-12
8	0.99	4.21e-07	1.01e-02	3.52e-11	1.27e-02
16	0.22	7.41e-06	1.31e+00	1.09e-08	6.09e+00
32	0.00	2.25e+00	1.31e+00	1.63e+01	7.47e+01

Таблица 2. Функция Растригина, $\alpha = 1$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	6.69e-10	1.46e-08	0.00e+00	3.92e-14
4	1.00	2.64e-08	1.55e-07	1.14e-13	9.43e-12
8	0.99	1.32e-06	6.45e-04	3.44e-10	8.02e-03
16	0.63	1.69e-05	3.48e-01	5.66e-08	2.34e+00
32	0.03	3.41e-04	2.52e+00	2.31e-05	4.68e+01

Таблица 3. Функция Растригина, $\alpha = 0.2$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	1.01e-09	1.78e-08	0.00e+00	5.83e-14
4	1.00	6.39e-09	1.19e-07	0.00e+00	4.12e-12
8	1.00	1.65e-06	8.22e-06	5.41e-10	1.64e-08
16	0.22	1.37e-05	1.11e-03	3.74e-08	6.68e-03
32	0.00	2.45e-05	2.91e-02	1.19e-07	6.59e-01

Таблица 4. Функция Розенброка, $\alpha = 6$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	4.03e-01	4.14e-01	0.00e+00	1.53e-06
4	0.94	6.57e-01	9.77e-01	1.07e-05	2.44e-03
8	0.05	2.95e-01	9.78e-01	7.17e-04	2.38e+00
16	0.00	5.16e-01	1.49e+00	5.64e-01	7.69e+01
32	0.00	1.48e+00	4.41e+00	1.39e+02	5.38e+03

Таблица 5. Функция Розенброка, $\alpha = 2$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	4.08e-01	4.14e-01	0.00e+00	7.19e-07
4	0.98	5.69e-01	9.80e-01	6.01e-06	1.73e-03
8	0.14	2.18e-01	1.59e+00	2.14e-05	2.79e-01
16	0.00	1.60e+00	2.83e+00	1.08e-02	7.61e+00
32	0.00	2.48e+00	4.25e+00	3.43e+00	2.01e+02

Таблица 6. Функция Розенброка, $\alpha = 0.2$

SN	p_{glob}	δX^*	$\overline{\delta X^*}$	δf^*	$\overline{\delta f^*}$
2	1.00	4.02e-01	4.14e-01	0.00e+00	7.67e-07
4	1.00	9.49e-01	1.00e+00	2.97e-07	1.33e-04
8	0.92	1.71e+00	1.82e+00	3.42e-05	3.08e-03
16	0.27	2.91e+00	3.00e+00	3.93e-04	1.47e-01
32	0.00	4.45e+00	4.64e+00	1.31e-01	6.98e+00

Графики зависимости вероятности локализации глобального минимума p_{glob} от размерности $|X|$ приведены на Рис. 3, 4.

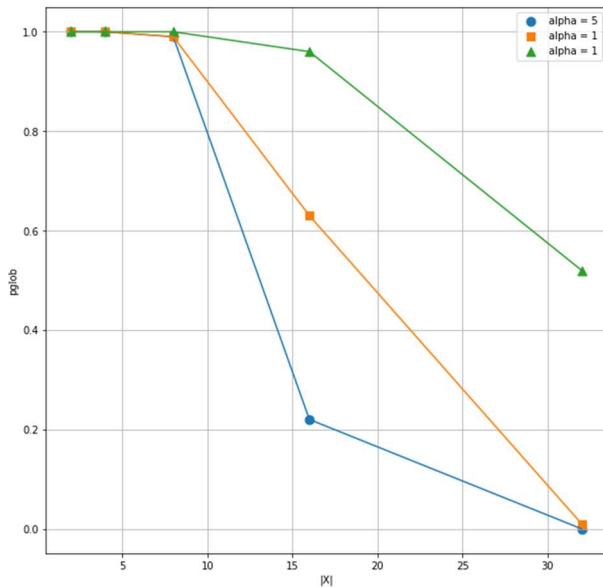


Рис. 3.

Зависимость p_{glob} от $|X|$ для ф-ии Растргина

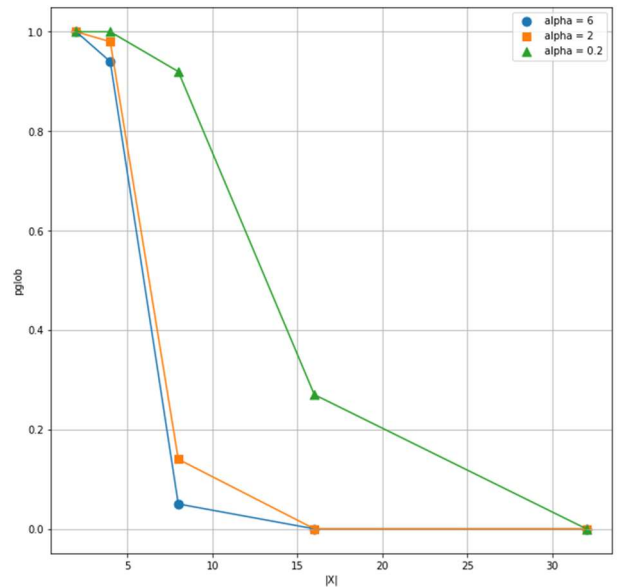


Рис. 4.

Зависимость p_{glob} от $|X|$ для ф-ии Розенброка

Графики погрешности целевой функции лучшего решения δf^* от размерности $|X|$ приведены на Рис. 5, 6. Стоит отметить, что для обоих алгоритмов при размерности $|X| = 2$ значения δf^* равны нулю. Т.к. шкала графика логарифмическая, возникают "скачки" графика вниз.

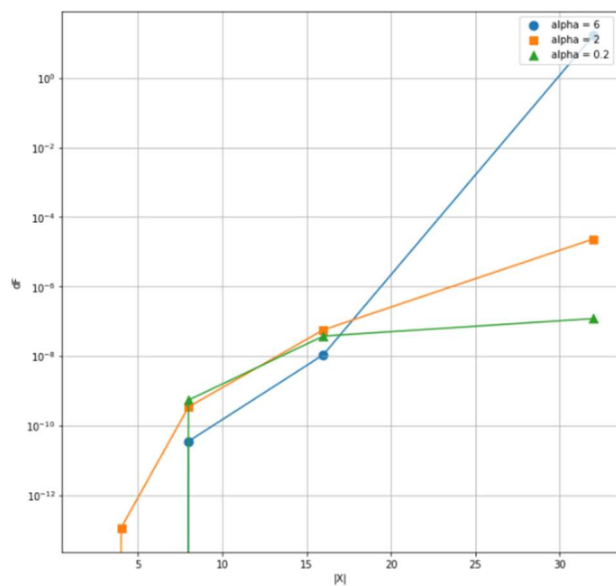


Рис. 5.
Зависимость δf^* от $|X|$ для ф-ии Растригина

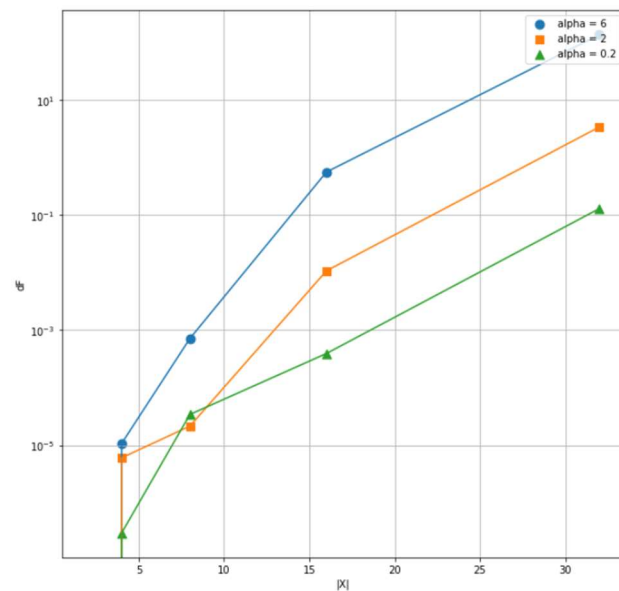


Рис. 6.
Зависимость δf^* от $|X|$ для ф-ии Розенброка

Графики средней погрешности целевой функции $\overline{\delta f^*}$ от размерности $|X|$ приведены на Рис. 7, 8.

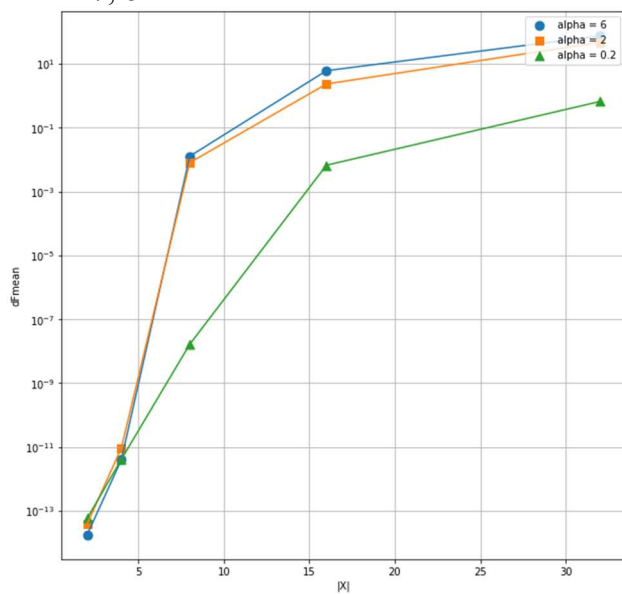


Рис. 7.
Зависимость $\overline{\delta f^*}$ от $|X|$ для ф-ии Растригина

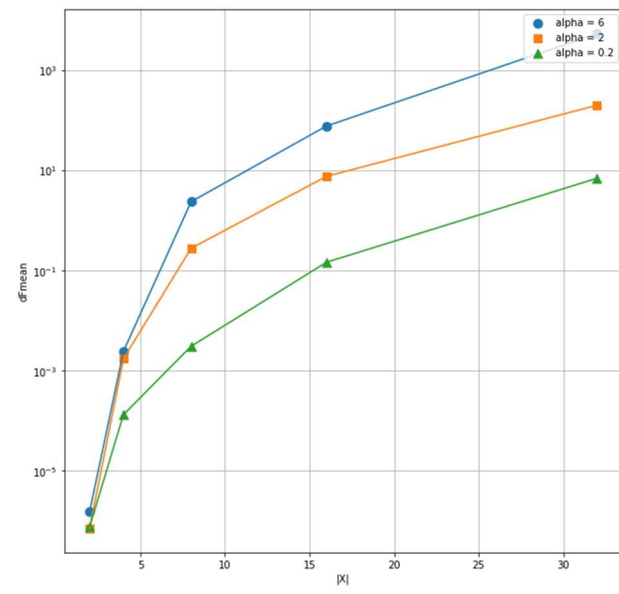


Рис. 8.
Зависимость $\overline{\delta f^*}$ от $|X|$ для ф-ии Розенброка

Графики нормы абсолютной погрешности по компонентам векторов лучшего решения δX^* от размерности $|X|$ приведены на Рис. 9, 10.

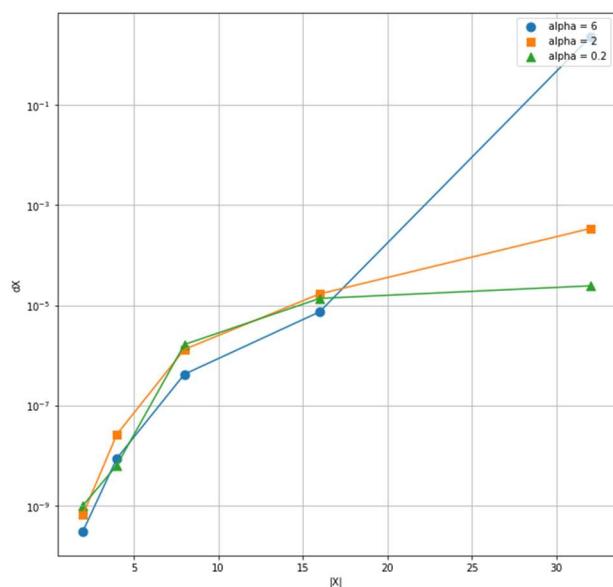


Рис. 9.
Зависимость δX^* от $|X|$ для ф-ии Растригина

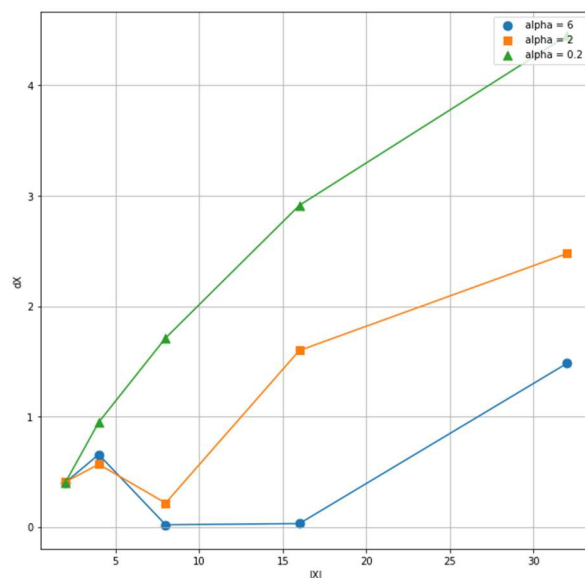


Рис. 10.
Зависимость δX^* от $|X|$ для ф-ии Розенброка

Графики нормы абсолютной погрешности по компонентам средних векторов $\overline{\delta X^*}$ от размерности $|X|$ приведены на Рис. 11, 12.

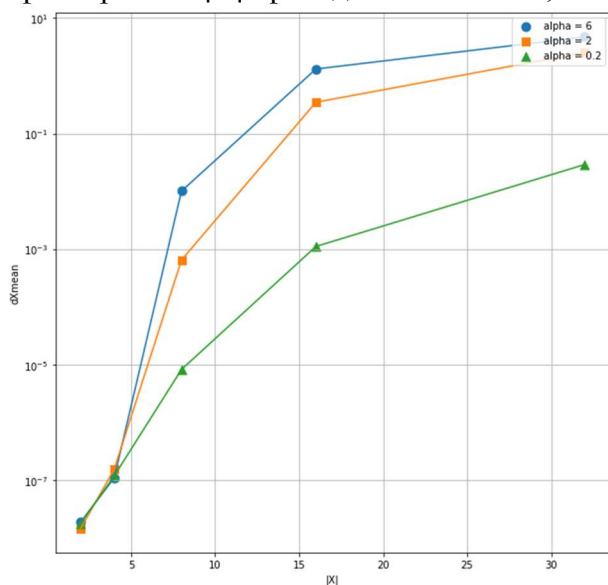


Рис. 10.
Зависимость $\overline{\delta X^*}$ от $|X|$ для ф-ии Растригина

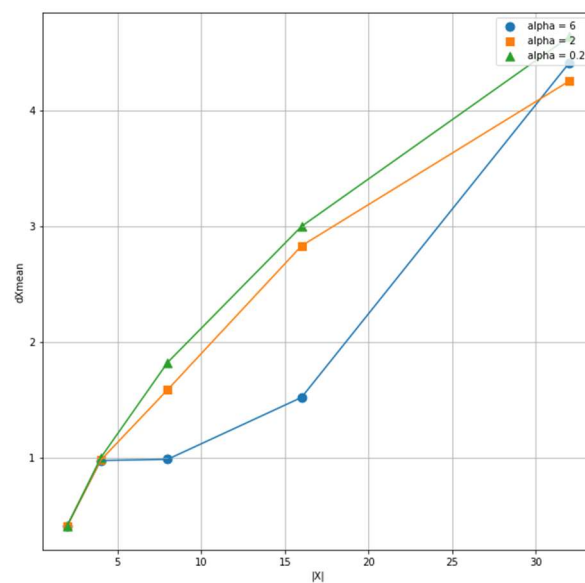


Рис. 11.
Зависимость $\overline{\delta X^*}$ от $|X|$ для ф-ии Розенброка

Заключение

Приведенный алгоритм практически гарантирует сходимость к глобальному минимуму для сложных целевых функций, таких как функции Растригина и Розенброка, при малых размерностях вектора X (до 8-и для функции Растригина; до 4-х для функции Розенброка), однако при увеличении размерности необходимо либо увеличение количества стартов алгоритма, либо уменьшение зоны начального распределения особей.

Также следует отметить, что алгоритм MNNABC может быть малоэффективен в случае, если сложность вычисления целевой функции сопоставима со сложностью вычисления евклидовых расстояний между особями популяции, в силу того, что количество вычислений такого расстояния на каждой итерации квадратично зависит от количества особей популяции, чего нет в исходном алгоритме ABC.

Использованная литература

1. Karaboga, D. (2005) *An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University*
2. K. Li, H. Wang, W. Wang, et al., *Improving artificial bee colony algorithm using modified nearest neighbor sequence*, *Journal of King Saud University – Computer and Information Sciences*, <https://doi.org/10.1016/j.jksuci.2021.10.012>