|  |  |
| --- | --- |
|  | **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 вектор действительных чисел размерностью ; скалярная целевая функция векторного аргумента . Задачей многомерной безусловной оптимизации называется задача поиска вектора , удовлетворяющего уравнению

**Описание алгоритма**

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

* попеременное использование на стадии *employeed bee* двух поисковый 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 ближайших соседей для особи , принадлежащей популяции мощностью . Приведем алгоритм построения такой последовательности, предложенный в [2]:

1. В качестве нулевого элемента последовательности берется исходная особь .
2. Из всех особей популяции выбираются множество особей  
   , т.е. все особи, имеющие целевую функцию меньшую, нежели .
3. В последовательность добавляется особь из множества , ближайшая к в смысле евклидова расстояния.
4. Пункты 2 и 3 циклически повторяются для последнего добавленного в последовательность элемента до тех пор, пока множество не пусто. После этого последовательность с количеством элементов 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 является лучшей в популяции, и её последовательность ближайших соседей состоят только из самой особи.

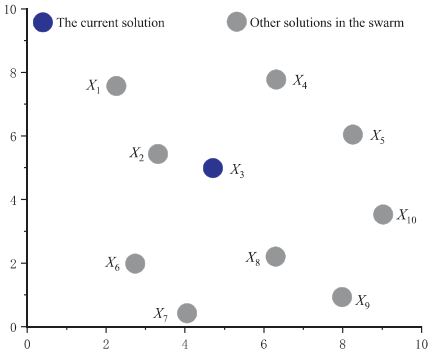


Рис. 1.1.

Выбор особи популяции для построения множества

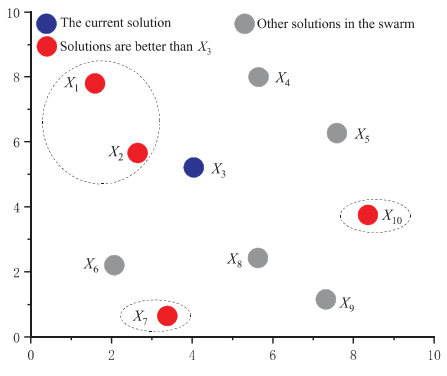


Рис. 1.2.

Построение множества

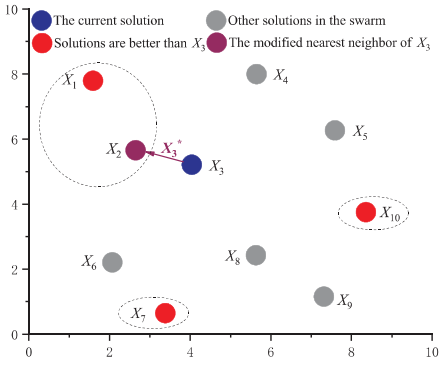


Рис. 1.3.

Выбор из множества ближайшего соседа

Как было сказано выше, последовательности используются в предложенных авторами стратегиях поиска, используемых для получения особи на следующей итерации алгоритма. Опишем первую из этих стратегий. Компоненты векторов находятся как

где , случайное целое число в интервале , случайное число в интервале , -я компонента лучшей особи популяции ,  
-я компонента вектора , случайно выбранного из всей популяции,  
 центр (вектор средних значений) последовательности , -я компонента , -я компонента вектора , принадлежащего последовательности .

В свою очередь, вторая стратегия описывается формулой

где -е компоненты соответственно особей , случайно выбранных из последовательности

Вместе эти две стратегии образуют пул стратегий , из которых выбирается стратегия для следующего обновления особи по принципу

Формула инициализация начальных положений особей популяции:

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

Алгоритм 1. Стадия employed bee

**for** each **do**

Сгенерировать по соответствующей стратегии ;

Вычислить ;

**if** < **then**

= ;

;

**else**

;

**end**

Обновить стратегию в соответствии с (4);

**end**

Алгоритм 2. Стадия onlooker bee

**for** to **do**

Получить последовательность для ;

Случайно выбрать из ;

Найти индекс в популяции для ( == );

Сгенерировать по соответствующей стратегии ;

Вычислить ;

**if** < **then**

= ;

;

**else**

;

**end**

Обновить стратегию в соответствии с (4);

**end**

Алгоритм 3. MNNABC

Инициализировать популяцию мощностью в соответствии с (5);

;

;

**while** **do**

Выполнить Алгоритм 1 (employeed bee);

Выполнить Алгоритм 2 (onlooker bee);

**if** **then**

Заменить новым в соответствии с (5) (scout bee);

;

**end**

**if** **then**

;

**else**

;

**end**

;

**end**

**Программная реализация**

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

Тестирование проводилось на сферической функции размерности с центром в точке (0, 0), критерием останова служила стагнация решения более 20-и итераций  
(т.е. ) при . Положения особей популяции на различных итерациях алгоритма приведены на Рис. 2.

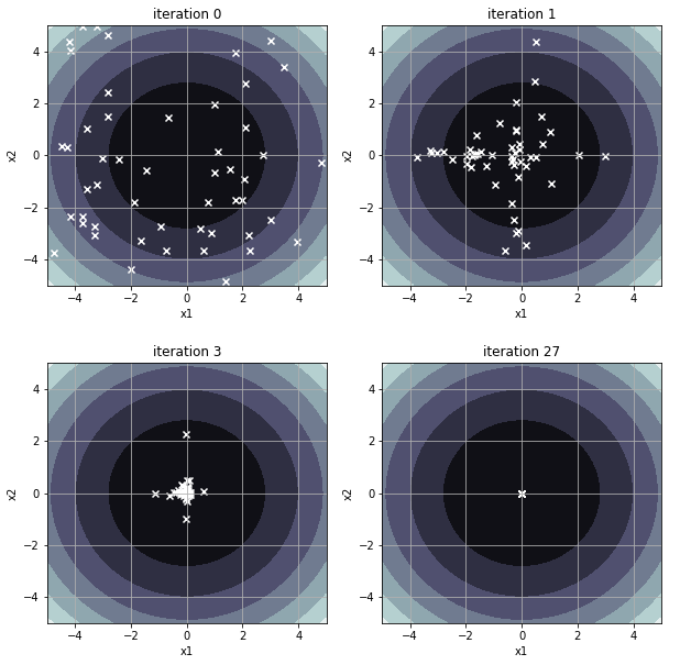


Рис. 2.

Положения особей на различных итерациях алгоритма

Найден минимум со значением целевой функции . Соответственные погрешности , .

**Вычислительный эксперимент**

Вычислительный эксперимента проводился для двух функций – Растригина, заданной уравнением

и Розенброка, заданной уравнением

для различных значений размерности .

При этом начальные распределения особей популяции задавались так, что для функции Растригина, и для функции Розенброка (поскольку её глобальный минимум расположен в точке ). Значения величины равны для функции Растригина и для функции Розенброка соответственно. Количество особей в популяции .

Критериями останова алгоритма служили стагнация вычислительного процесса на протяжении 20 итераций при .

Для каждой размерности проводились мультистарты в количестве 100. Метриками качества работы алгоритма служили абсолютная погрешность по лучшему значению целевой функции , нормы абсолютная погрешность по компонентам вектора лучшего решения , а также средние значения абсолютных погрешностей по целевой функции и норме вектора , а также вероятность локализации глобального минимума . Критерием локализации глобального минимума служит .

Результаты вычислений приведены в Табл. 1-6.

Таблица 1. Функция Растригина,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| *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 от размерности приведены на Рис. 3, 4.

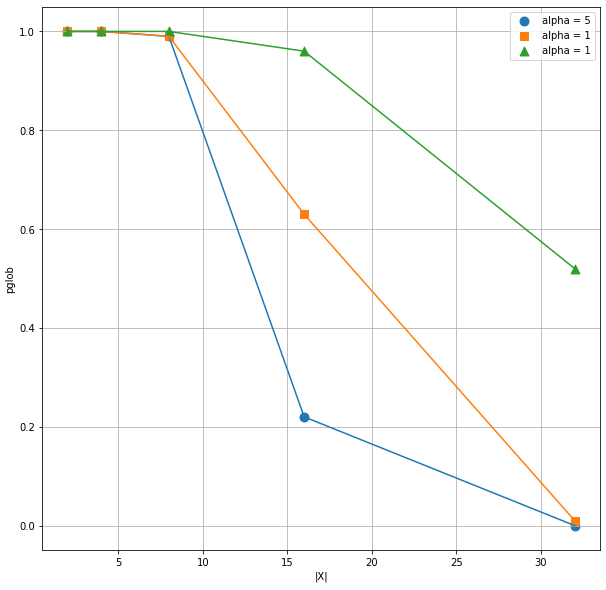


Рис. 3.

Зависимость от для ф-ии Растригина

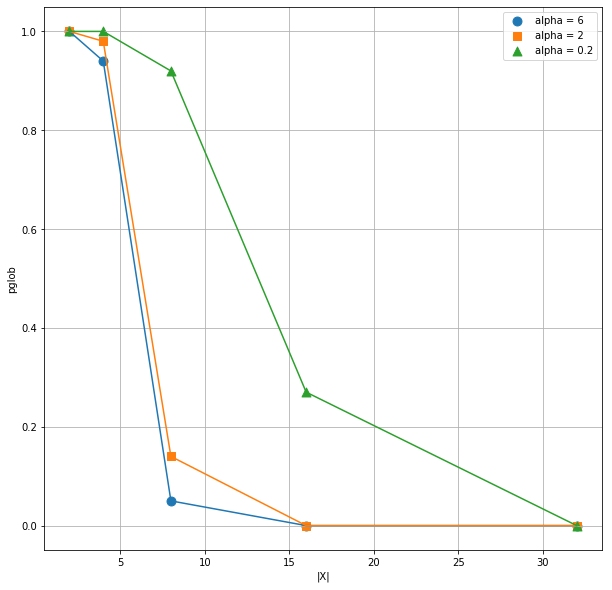


Рис. 4.

Зависимость от для ф-ии Розенброка

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

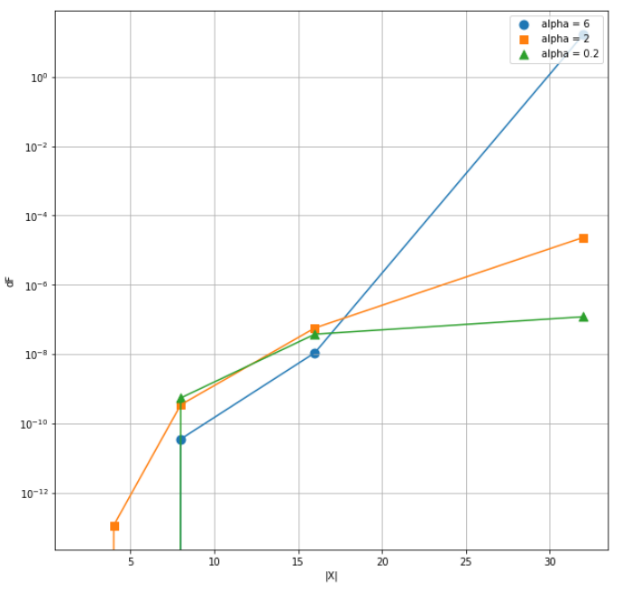


Рис. 5.

Зависимость от для ф-ии Растригина

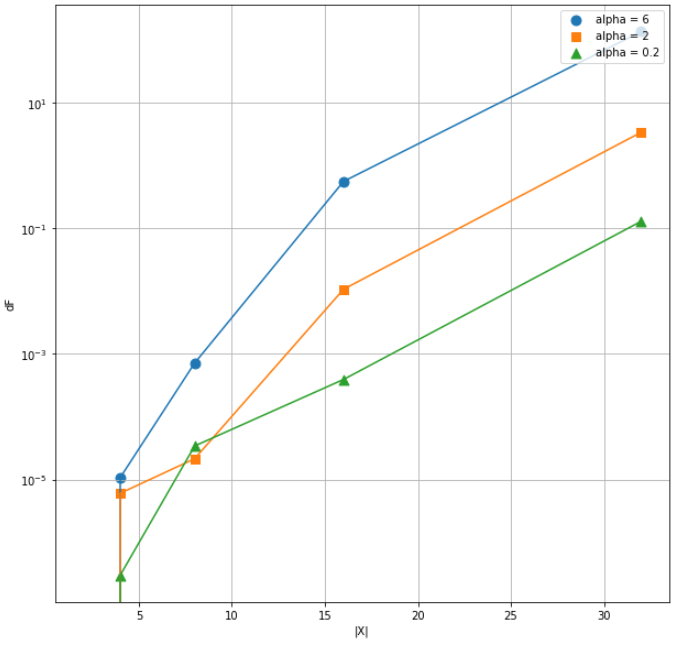


Рис. 6.

Зависимость от для ф-ии Розенброка

Графики средней погрешности целевой функции от размерности приведены на Рис. 7, 8.

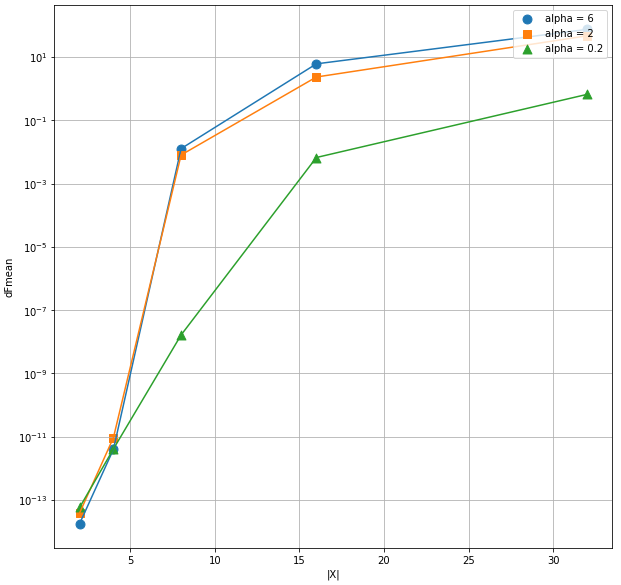


Рис. 7.

Зависимость от для ф-ии Растригина

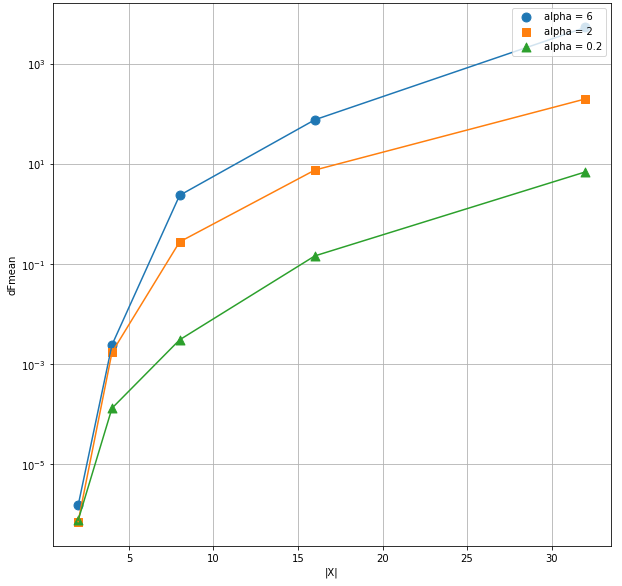


Рис. 8.

Зависимость от для ф-ии Розенброка

Графики нормы абсолютной погрешности по компонентам векторов лучшего решения от размерности приведены на Рис. 9, 10.

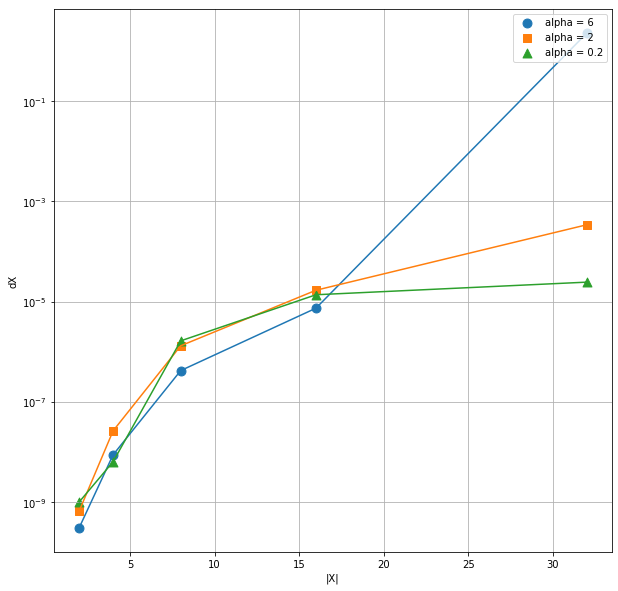


Рис. 9.

Зависимость от для ф-ии Растригина

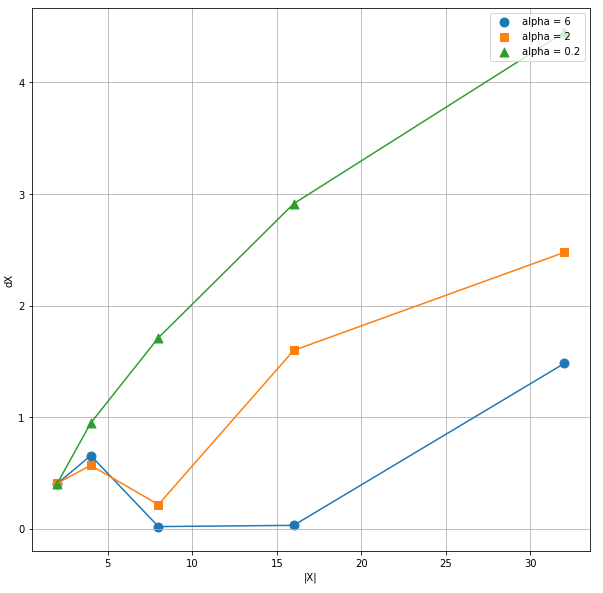


Рис. 10.

Зависимость от для ф-ии Розенброка

Графики нормы абсолютной погрешности по компонентам средних векторов от размерности приведены на Рис. 11, 12.

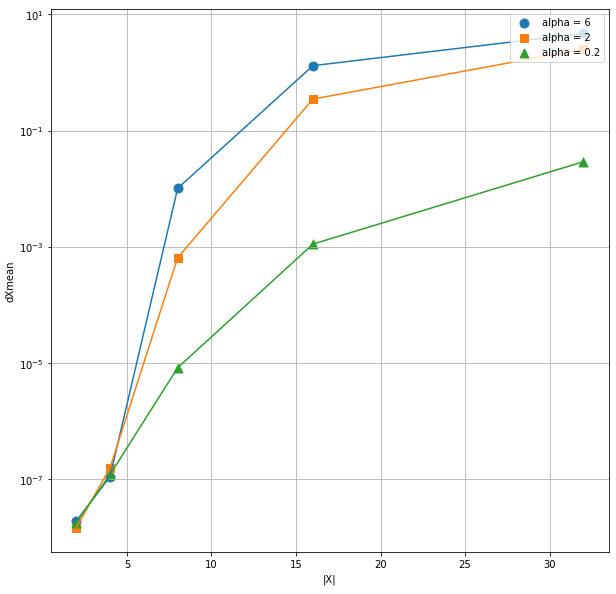


Рис. 10.

Зависимость от для ф-ии Растригина

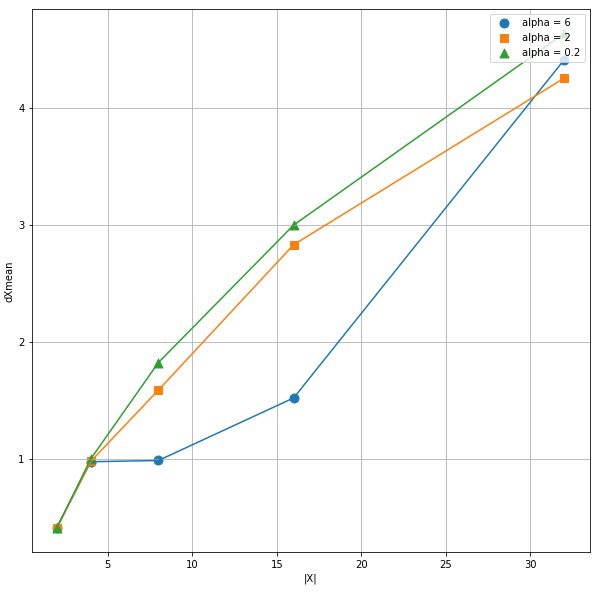


Рис. 11.

Зависимость от 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 (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*](https://doi.org/10.1016/j.jksuci.2021.10.012)