Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 1532"

**Сравнение эффективности алгоритмов оптимизации на базе искусственного интеллекта на тестовых задачах вещественной оптимизации**

10 класс, ГБОУ Школа №1532,

Исупов Матвей Викторович

Руководитель: учитель информатики, ГБОУ Школа №1532,

Сергиенко Антон Борисович

Москва 2023

Содержание

[Введение 3](#_Toc128253653)

[Гипотеза 3](#_Toc128253654)

[Цель и задачи работы 3](#_Toc128253655)

[Методика выполнения работы 4](#_Toc128253656)

[Генетический алгоритм 4](#_Toc128253657)

[Модификации генетического алгоритма 6](#_Toc128253658)

[Критерий Вил-Коксона 9](#_Toc128253659)

[Метод роя частиц 10](#_Toc128253660)

[Модификации PSO 12](#_Toc128253661)

[Сравнение алгоритмов оптимизации 14](#_Toc128253662)

[Вывод 16](#_Toc128253663)

[Список используемой литературы 17](#_Toc128253664)

# Введение

В последние годы интерес к такому направлению как Natural Computing сильно возрос. Ученые выделяют три класса:

1. Черпание вдохновения из природных процессов.
2. Использование программ для симуляции, синтеза природных явлений.
3. Использование природных молекул для вычисления.

Основными областями исследований, которые составляют эти три класса, являются: искусственные нейронные сети, эволюционные алгоритмы, роевой интеллект (роевой метод частиц), искусственные иммунные системы, фрактальная геометрия, искусственная жизнь, вычисления ДНК и квантовые вычисления.

Стоит упомянуть, что проблема оптимизации сложных систем – одна из главный проблем в современном направлении нейронных сетей и различных алгоритмов глобальной оптимизации. Задачи, где применяются подобные исследования варьируются от бизнес прогнозов и социальных исследований, до различных комбинаторных задач.

Все вышеперечисленные методы являются способами решения задач глобальной оптимизации, в свою очередь, мой проект является исследованием в области сравнения подобных алгоритмов глобальной оптимизации на тестовых задачах.

# Гипотеза

Моя гипотеза заключается в следующем: популяция лучше приспосабливается с течением поколений, ведь вопрос: найдет ли толпа решение – имеет очень непредсказуемый ответ.

# Цель и задачи работы

Анализ эффективности алгоритмов глобальной оптимизации на тестовых функциях и разработка их модификаций с целью повышения эффективности.

В ходе разработки дорожной карты проекта были поставлены следующие задачи:

1. Реализация стандартного генетического алгоритма на языке Python.
2. Адаптирование ген. алгоритма под решение проблемы вычисления локальных экстремумов функции.
3. Получение решения и значений.
4. Реализация метода роя частит на языке Python.
5. Исследование различных модификаций обоих алгоритмов с целью повышения эффективности решения данной задачи.
6. Сравнение работы двух алгоритмов глобальной оптимизации.

# Методика выполнения работы

## **Генетический алгоритм**

Генетический алгоритм, известный как GA или ГА, представляет собой метод поиска, используемый для решения задач глобальной оптимизации. Он основан на механизмах, аналогичных естественному отбору в природе, и является разновидностью эволюционных вычислений. Этот метод использует методы эволюционного отбора, такие как наследование, мутации, отбор и кроссинговер, чтобы достичь лучшей оптимизации.

Для решения задачи нахождения локальных экстремумов функции используется алгоритм, основанный на классическом генетическом алгоритме Джона Холланда. Работу алгоритма можно разбить на несколько этапов:

1. генерация первого поколения;
2. отбор родителей;
3. скрещивание;
4. мутации;
5. создание нового поколения.

Пункты с 2-5 выполняются до тех пор, пока не пройдёт заданное количество поколений.

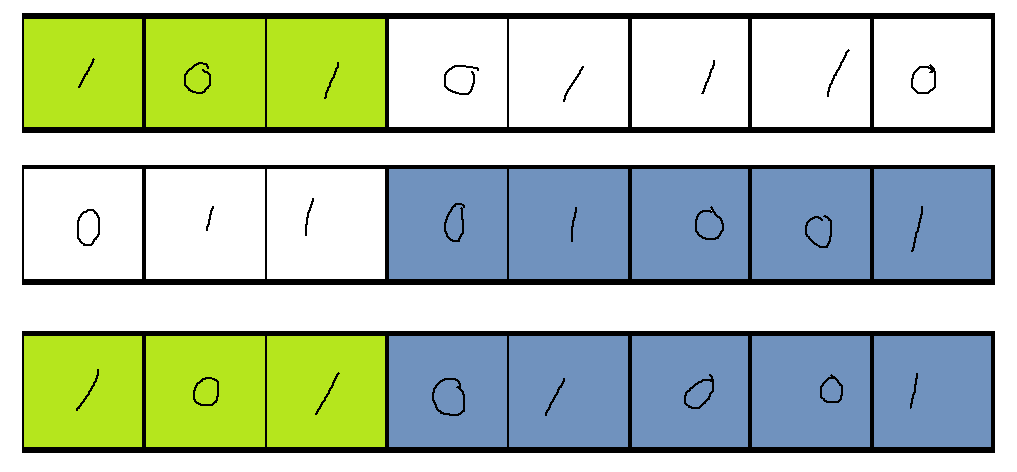
 В задаче НГЭФ (нахождения глобального экстремума функции) в качестве индивида берётся бинарная строчка с задаваемой длиной. Характеристикой сравнения, в свою очередь, является количество единиц в заданном индивиде. Популяцией является массив с N-задаваемым количеством элементов.

Рисунок 1 – Бинарная модель генотипов в популяции

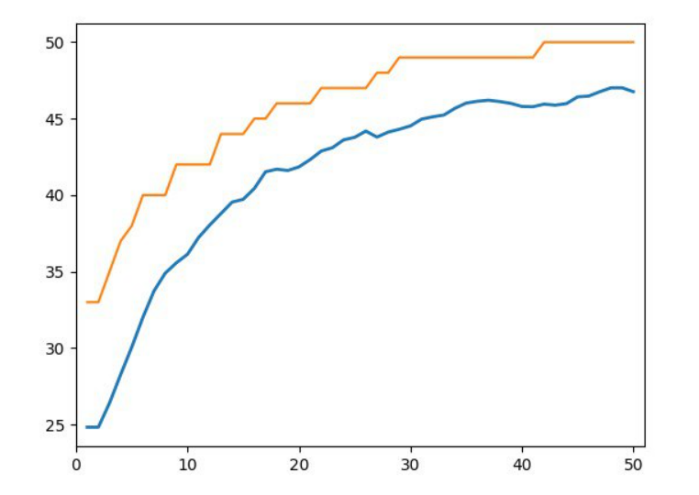
 Рождение индивидов следующего поколения происходит следующим образом: берутся несколько случайных индивидов в популяции, происходит сравнение строк, отбираются наилучшие (считаются те, у кого единиц в бинарной строчке больше) представители в качестве родителей для скрещивания. Затем используется метод деления обоих родителей на части (одна часть от первого родителя, вторая – остаток от второго родителя. (рис. 1)

Рисунок 2 – значение средней пригодности к приг. лучшего индивида

Таким образом собирается новое поколение, такого же размера как предыдущее и заменяет его. В большинстве случаев самые качественные индивиды передают свои гены из поколения в поколения, за счёт чего каждый раз качество индивидов из поколения в поколение улучшается (рис. 2),

где

* x – качество индивидов;
* y – N-поколение;
* оранжевая линяя – качество лучшего индивида в поколении;
* голубая линия – среднее значение всех индивидов в поколени.

Количество индивидов в одном поколении, количество поколений, шанс мутации и ограничения функции – настраиваемые параметры генетического алгоритма.

В функции fitness в ГА координаты x и y задаются следующей формулой:

### **Модификации генетического алгоритма**

В процессе доработки генетического алгоритма было решено продумать некоторые возможные модификации, что способны улучшить его качество. Для начала я решил провести эксперимент с одним из главных параметров ГА – вероятностью мутации.

Первичная идея, заключающаяся в обычном уменьшении вероятности с итерациями показала себя недееспособной, после чего было решено добавить некоторую зависимость в формулу перерасчёта вер. мутации, что можно описать следующей формулой:

где

* – среднее арифметическое;
* – задаваемое значение вероятности.

За основу был взят следующий ход рассуждений: ‘Эволюционные методы приспособления работают хуже, покуда популяция не выходит за рамки пригодности.’

Значение 0.05 – константа, что была выведена в результате проведенных опытов.

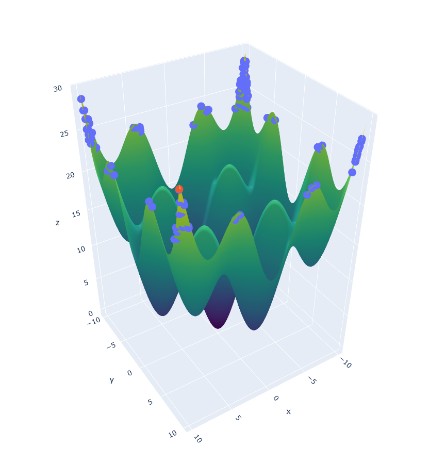
В качестве оптимизируемой функции была взята следующая зависимость:

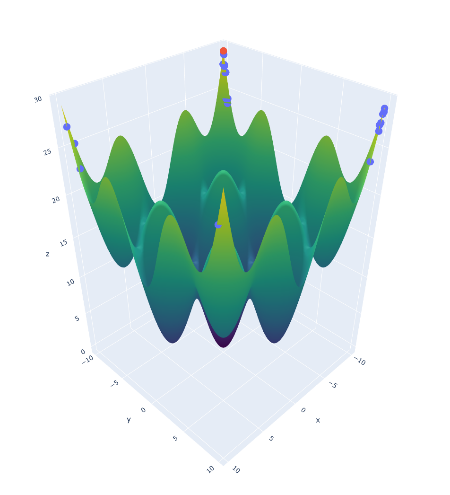
Исходные параметры обоих вариаций:

* Количество индивидов – 100;
* Длина индивидов – 32;
* Количество итераций – 500;
* Вероятность мутации – 0.1;
* Lim1 – -10;
* Lim2 – 10;
* max

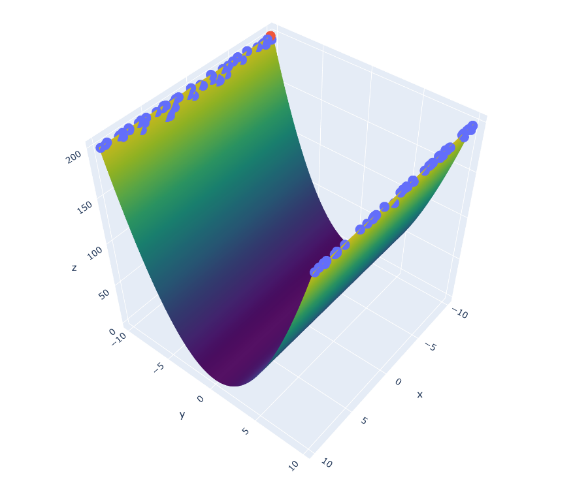
|  |  |
| --- | --- |
| Модификация | Классический алгоритм |
| -0.0022425534980673057 | -0.0006717369065685338 |
| -9.52306394407384e-05 | -0.0006830993887316342 |
| -8.126037448610686e-05 | -0.00024145274596588202 |
| -4.083974121737291e-05 | -0.002483959676483339 |
| -0.00012317116935000151 | -0.0003371956284549569 |

Таблица 2 – сравнение значений модифицированного и классического ГА

Рисунок 3.1 – пример работы обычного алгоритма

Рисунок 3.2 – пример работы модифицированного алгоритма

Также, модифицированный алгоритм показал себя лучше и на прочих тестовых функциях:

Рисунок 4.1 – пример работы обычного алгоритма на иной тестовой функции

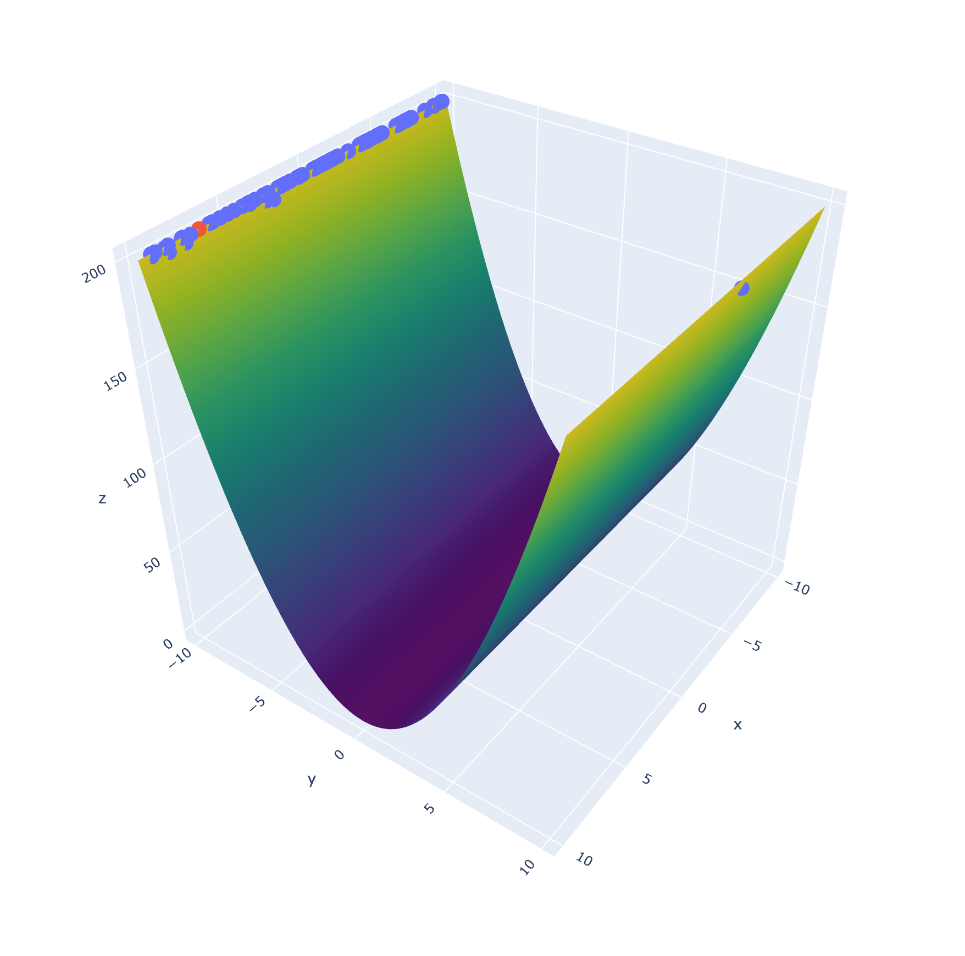


Рисунок 4.2 – пример работы модифицированного алгоритма на иной тестовой функции

В результате проведённых исследований для сравнения алгоритмов было решено использовать критерий Вил-Коксона.

### **Критерий Вил-Коксона**

T-критерий или критерий Вил-Коксона – статистический тест, используемый для проверки различий между двумя выборками парных или независимых измерений по уровню какого-либо количественного признака.

Его алгоритм заключается в:

1. Формировании массивов из двух выборок;
2. Сортировании массивов в порядке возрастания;
3. Обозначении рангов элементов;
4. Высчитывании значения статистики;
5. Утверждения уровня значимости;
6. Подсчёте критических границ статистики.

На мой взгляд, намного целесообразнее начать подробное описание критерия стоит с четвёртого пункта.

Проведя все необходимые процедуры с дата сетом, был получено значение статистики, равное 36, после чего необходимо было выбрать уровень значимости , в промежуток границ которого входит значение статистики. В моём случае таковым оказалось значение .

Далее, необходимо высчитать критические границы и , что высчитываются в соответствии с ниже представленными формулами:

где

* – значение обратной функции нормального распределения с параметрами (0, 1);
* – длины векторов двух выборок.

В моём случае соответственно равные: = 20; = 35, после чего необходимо проверить справедливость следующей формулы:

,

И если значение функции не находится в пределах заданного вектора - можем сделать вывод: при уровне значимости 𝑄, выборки неоднородны по критерию Вил-Коксона, из чего следует факт превосходства модифицированного генетического алгоритма над классическим.

## **Метод роя частиц**

Метод роя частиц (МРЧ, PSO или ПСО) — метод численной оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции.

Идея алгоритма была частично заимствована и адаптирована психологами Кеннеди и Эберхардом, которые занимались исследованием поведения скоплений животных. Модель была немного упрощена и добавлены элементы поведения толпы людей, поэтому, индивиды (ГА) или элементы были названы частицами. Он решает проблему, имея популяцию возможных решений – частиц, перемещая которые в пространстве поиска в соответствии с простой математической формулой относительно положения и скорости частицы.

Работу алгоритма можно разбить на следующие этапы:

1. Создание роя.
2. Нахождение лучшего решения для каждой частицы.
3. Нахождения лучшего среди всех частиц.
4. Коррекция скорости каждой частицы.
5. Перемещение частицы.
6. Вывод результата.

Пункты с 2-5 выполняются до тех пор, пока не пройдёт заданное количество итераций или условие прекращения алгоритма не будет выполнено.

Одна из главных особенностей PSO (particle swarm optimization) является уравнение векторной скорости. Что задаётся формулой на рис. 3,

где

* – коэффициент личного вектора скорости;
* – коэффициент общественного вектора скорости;
* , – случайные коэффициенты в промежутке [0, 1];
* – коэффициент инерции;
* – предыдущее значение скорости;
* – лучшее значение индивида;
* – лучшее значение роя;
* - значение (t) точки в (i) итерации.

### **Модификации PSO**

В моей реализации метода роя частиц присутствует модификация, основанная на изменение личностного, глобального и инерциального коэффициентов в уравнении векторной скорости (, , ).

Данные коэффициенты изменяются по следующим формулам:

Подобная модификация используется для того, чтобы в начальных итерациях расширить зону поиска с большим коэффициентом , а затем, когда одна из точек обнаружит глобальный экстремум, рой начал стремиться к этой точке при увеличении коэффициента . Также необходимо уменьшать коэффициент инерции для поиска наиболее приближённого значения к глобальному экстремуму.

В качестве оптимизируемой функции была взята следующая зависимость:

Исходные параметры обеих вариаций:

* Количество индивидов – 32;
* Количество итераций – 100;
* Lim1 – -10;
* Lim2 – 10;
* max.

|  |  |
| --- | --- |
| Модификация | Классический алгоритм |
| 197.964940448349 | 187.00941142755548 |
| 197.30952377944215 | 192.67747220008312 |
| 199.12183640000237 | 194.02478678041484 |
| 197.18410821114276 | 156.84365767141742 |
| 198.32082127639313 | 192.1959597676409 |

Таблица 1 – сравнение эффективности PSO и мод. PSO

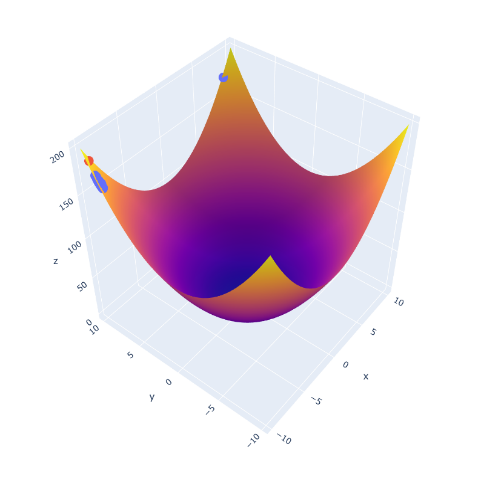
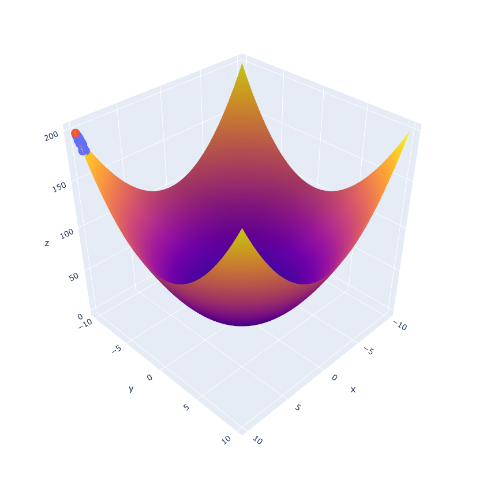
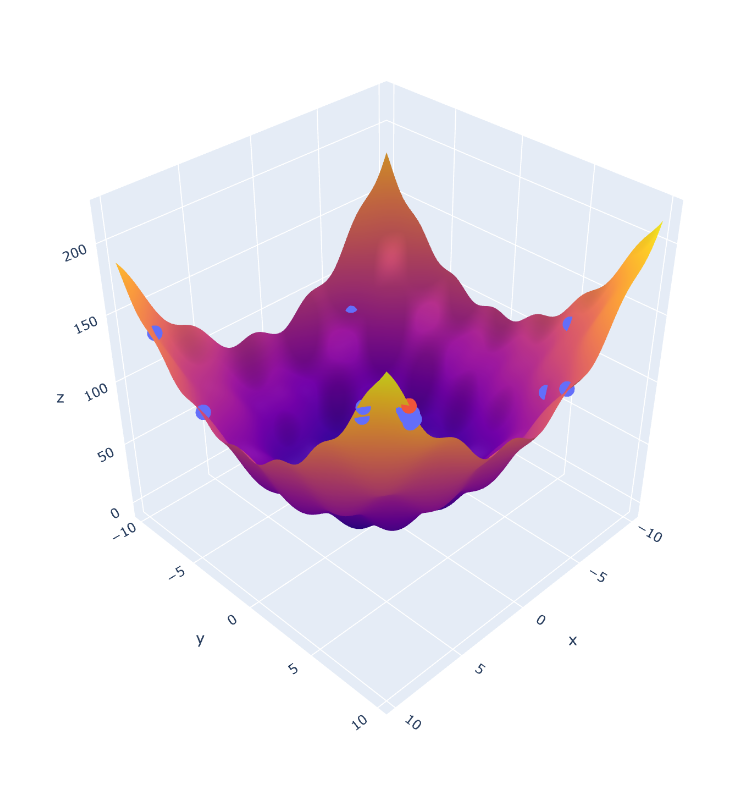
По критерию Вил-Коксона, при уровне значимости Q = 0.02 выборки неоднородны – модификация улучшила показатели.

Рисунок 3.1 – результат работы классического алгоритма

Рисунок 3.2 – результат работы модифицированного алгоритма

Также, модифицированный алгоритм показал себя лучше и на прочих тестовых функциях:



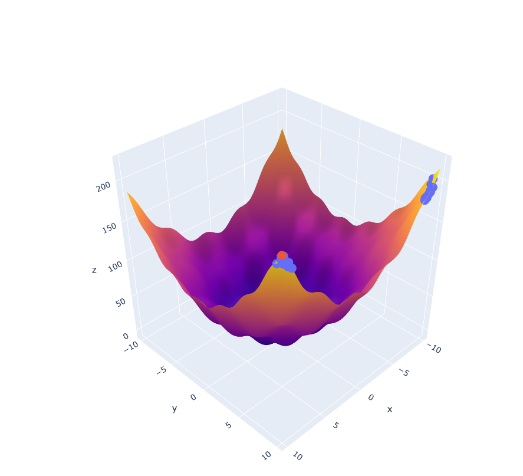
Рисунок 4.1 – пример работы обычного алгоритма на иной тестовой функции

Рисунок 4.2 – пример работы модифицированного алгоритма на иной тестовой функции

## **Сравнение алгоритмов оптимизации**

В качестве функции, которую необходимо оптимизировать, мы возьмём следующую формулу:

Исходные параметры ГА (мод.):

* Количество индивидов – 100;
* Длина индивидов – 32;
* Количество итераций – 100;
* Вероятность мутации – 0.2;
* Lim1 – -3;
* Lim2 – 13;
* max.

Исходные параметры PSO (мод.):

* Количество индивидов – 100;
* Количество итераций – 100;
* Lim1 – -3;
* Lim2 – 13;
* max.

Таблица ниже – сравнение работы двух алгоритмов оптимизации.

|  |  |
| --- | --- |
| GA | PSO |
| 46.28978566816091 | 46.28902391955722 |
| 46.28978542327448 | 46.28977479436523 |
| 46.289786206352694 | 46.289520346678664 |
| 46.28978618209494 | 46.28940098740459 |
| 46.289786140944 | 46.289783951435794 |
| 46.28978618209494 | 46.28855562531273 |
| 46.289785377914534 | 46.28951605555745 |
| 46.28978598589301 | 46.2897733094083 |
| 46.289786206352694 | 46.28968562248354 |

Таблица 3 – сравнение GA и PSO алгоритмов.

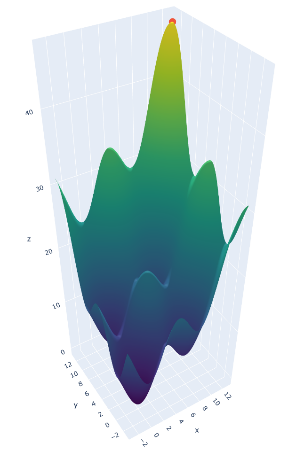
По критерию Вил-Коксона выборки неоднородны при любом уровне значимости, что свидетельствует о полной доминации генетического алгоритма над методом роя частиц.

Рисунок 5 – тестовая функция для PSO и GA

Также, хотел бы уточнить, что вычисления для PSO алгоритма происходят 8-12 сек. в отличие от GA, работа которого занимает не более 5 сек.

# **Вывод**

Автором были решены все поставленные задачи, приведены сравнительные анализы эффективности алгоритмов глобальной оптимизации, а также разработаны эффективные модификации этих алгоритмов, в чем и заключается научная новизна данной работы.

Подводя итоги, хотелось бы уточнить, что эволюционные механизмы отлично приспосабливаются с увеличением количества итераций, в отличие от так называемых “социальных” алгоритмов. Ведь с течением времени популяция приспосабливается гораздо лучше.

# Список используемой литературы

1. Разработка простого генетического алгоритма [Internet] http://www.wikiznanie.ru/ru-wz/index.php/Разработка\_простого\_генетического\_алгоритма
2. Cooperative and Adaptive Algorithms [Internet]

<https://uwaterloo.ca/scholar/sites/ca.scholar/files/ahilal/files/lecture-6-1.pdf>

1. Сергиенко, А. Б. Критерий Вилкоксона W для проверки однородности выборок. v. 1.2