



## Аннотация

В настоящее время генетические алгоритмы являются перспективным направлением в области искусственного интеллекта. С их помощью можно решать сложные практические задачи глобальной оптимизации. Одним из примеров подобных задач является «Проблема треугольников Хейльбронна» (Heilbronn triangle problem), являющаяся удобным полигоном для тестирования авторских модификаций генетического алгоритма, которые могут быть использованы для решения иных задач оптимизации.

Автором была рассмотрена расширенная задача треугольников Хейльбронна в трехмерном пространстве.

Также был предложен и реализован алгоритм, в котором было осуществлено решение задачи, а также проведено исследование эффективности различных алгоритмов с получением рекомендаций по использованию генетических алгоритмов для решения задач глобальной оптимизации. Вариант генетического алгоритма с модификациями, предложенными автором, для решения задачи показал свою эффективность.

## Содержание:

1	Проблема треугольников Хейльбронна .....	6
1.1	Формулировка задачи .....	6
1.2	История .....	6
1.3	Лучшие известные результаты .....	6
1.4	Вариации данной проблемы .....	7
2	Генетический Алгоритм .....	9
2.1	Определение .....	9
2.2	Работа генетического алгоритма .....	9
2.3	Проверка правильности работы генетического алгоритма.....	12
2.4	Реализованные дополнения к стандартному генетическому алгоритму.....	12
2.4.1	Методы селекции .....	13
2.4.2	Методы кроссинговера .....	15
2.4.3	Вещественное представления хромосомы генетического алгоритма.....	16
3	Статистика.....	18
3.1	Введение .....	18
3.2	Сравнение методов мутации бинарного генетического алгоритма .....	18
3.3	Сравнение методов селекции бинарного генетического алгоритма.....	19
3.4	Сравнение методов кроссинговера бинарного генетического алгоритма.....	20
3.5	Статистика методов селекции вещественного генетического алгоритма. ....	21
3.6	Статистика методов кроссинговера вещественного генетического алгоритма.....	21
3.7	Выводы по статистике .....	22
4	Модификации .....	24
4.1	Сборка оптимального генетического алгоритма .....	24
4.2	Модификации к оптимальным генетическим алгоритмам .....	24
5	Симбиотические модификации.....	26
5.1	Введение .....	26
5.2	Статистика .....	26
5.3	Сборка симбиоза .....	27
6	Треугольники Хейльбронна .....	29
6.1	Треугольники Хейльбронна в плоскости .....	29
6.2	Треугольники Хейльбронна в пространстве .....	30
	Вывод:.....	31
	Список используемой литературы:.....	33

## Введение

Проблема треугольников Хейльбронна (Heilbronn triangle problem) в дискретной теории геометрии является проблемой размещения точек в плоскости и пространстве, когда нужно найти такое расположение  $n$  точек в единичном квадрате, чтобы площадь треугольника с минимальной площадью среди всех возможных треугольников из этих точек была максимальной (Рисунок 1). Методики, использованные для решения данной задачи, могут помочь нам в исследовании генетического алгоритма, который позволяет осуществлять поиск оптимальных решений практических задач.

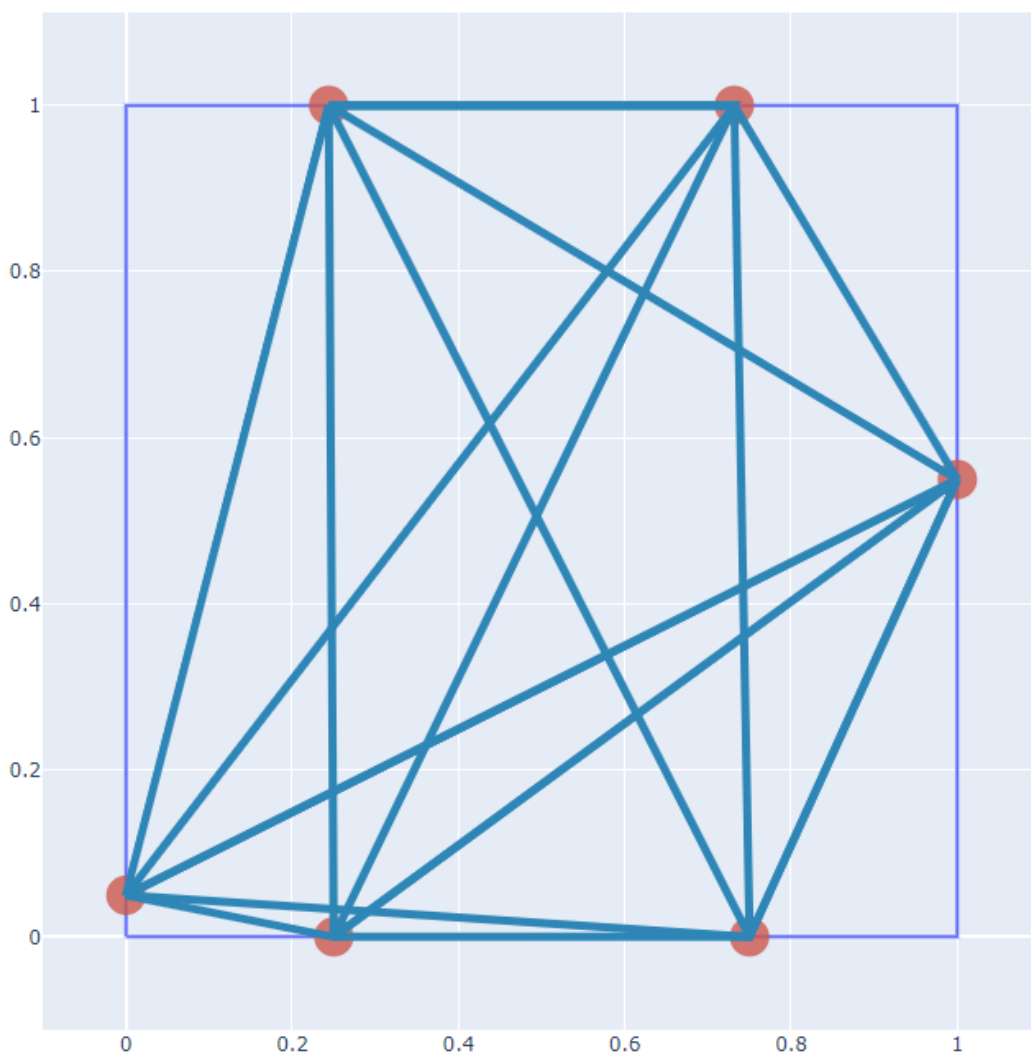


Рисунок 1 – Треугольники Хейльбронна (6 точек)

Для решения данной задачи автором было проведено исследование эффективности различных алгоритмов с получением рекомендаций по использованию генетических алгоритмов для решения задач глобальной оптимизации, а также предложена и реализована программа на языке C++, в которой было осуществлено решение данной задачи.

Актуальность работы.

Данная задача является **нерешенной** на данный момент и способ решения задач глобальной оптимизации путем использования генетических алгоритмов является актуальным и современным, а сами генетические алгоритмы являются перспективным направлением в области искусственного интеллекта, работа которых еще не изучена до конца, поэтому эмпирические и статистические данные, полученные в работе, могут потребоваться в других исследованиях на эту тему.

Теоретическая новизна работы.

Проблема треугольников Хейльбронна малоизвестна в русскоязычных кругах. Также закономерности и значения для трехмерного пространства не были еще полностью изучены, а поиск решений путем использования генетических алгоритмов является не самым популярным. Автором был предложен эффективный генетический алгоритм и его модификаций для поиска значений треугольников Хейльбронна в трехмерном пространстве.

# 1 Проблема треугольников Хейльбронна

## 1.1 Формулировка задачи

Даны  $n$  точек в площади единичного квадрата, необходимо найти максимальную возможную площадь минимального по площади треугольника.

Однако, самая первая формулировка включала в себя не площадь единичного квадрата, а площадь замкнутой выпуклой фигуры, но позже математиками было доказано, что оптимальные значения для каждой фигуры отличаются лишь постоянным множителем и не зависят от формы, поэтому для упрощения поиска была взята площадь единичного квадрата. [6]

## 1.2 История

Названа данная задача в честь немецкого математика Ганса Хейльбронна, который ее сформулировал в 1950-х, а также он дал гипотезу, что наименьшая максимальная площадь обратно пропорциональна к квадрату числа точек. Гипотеза оказалась ложной, но асимптотическая скорость роста площадей треугольников остается неизвестной. Более того, оптимальные решения неизвестны даже для маленького количества точек, а универсального математического метода решения нет. [9]

## 1.3 Лучшие известные результаты

Лучшие известные результаты были найдены для конфигураций до 16 точек (Рисунок 2, Таблица 1) Майлом Голдбергом в 1972 году. Конструкции Голдберга для шести точек на границе квадрата размещаются так, чтобы образовать аффинное преобразование вершин правильного многоугольника. Для больших значений Франческ Комеллас и Льюис Йебра в 2002 году улучшили оценки Голдберга, после чего конфигурации до 7 точек были признаны оптимальными. [8]

Дальнейшие исследования генетических алгоритмов для решения данной задачи, опираются именно на эти результаты. [1]

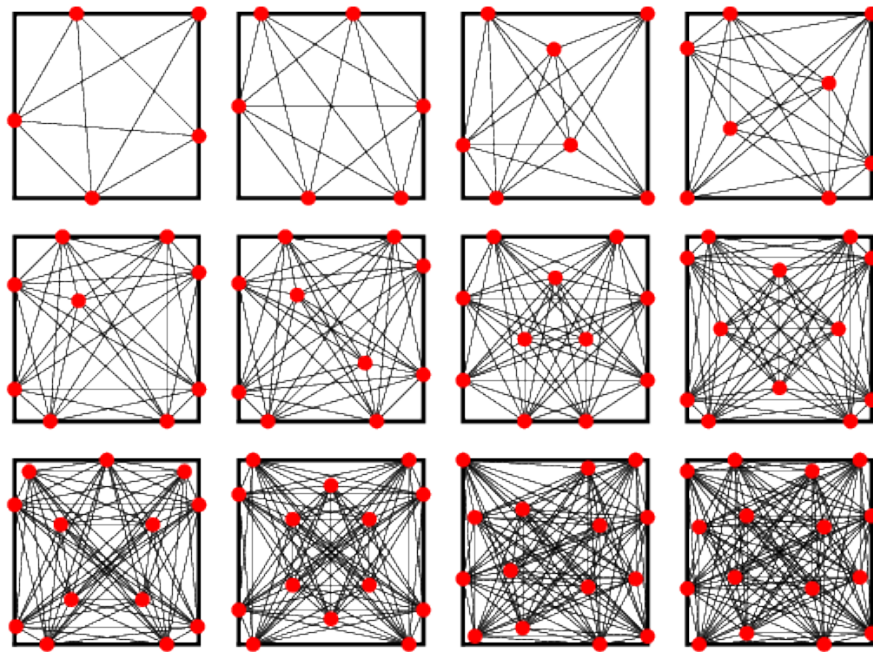


Рисунок 2 – Лучшие известные конфигурации точек от 3 до 16 [1]

$n$ - точек	площадь	$n$ - точек	площадь
3	0.5	10	0.046537...
4	0.5	11	0.037037...
5	0.1924...	12	0.032599...
6	0.125	13	0.026697...
7	0.083859...	14	0.024304...
8	0.072356...	15	0.020789...
9	0.054876...	16	0.020528...

Таблица 1 – Лучшие известные результаты от 3 до 16 точек [1]

#### 1.4 Вариации данной проблемы

Проблема треугольников Хейльбронна была перенесена из двумерного пространства в  $d$ -мерное пространство Гиллом Баркетом. Далее Баркет и его коллега Алина Шайхет исследовали закономерности данной проблемы для трехмерного пространства. [5]

Автором приведен вариант решения задачи в трехмерном пространстве с использованием генетического алгоритма.



## 2 Генетический Алгоритм

### 2.1 Определение

Генетический алгоритм – это эвристический алгоритм поиска, вдохновленный процессом природной селекции и принадлежащий к большому классу эволюционных алгоритмов. [9]

Генетические алгоритмы могут отличаться по многим параметрам. Одной из ключевых идей является представление потенциальных решений в виде “хромосом” (индивидов). Существует два наиболее часто используемых представления: вещественное и в виде бинарной строки. Все эти два представления будут использованы для исследования эффективности.

### 2.2 Работа генетического алгоритма

Генетический алгоритм работает на семи основных этапах (Рисунок 3). Первоначально автором было реализовано стандартное представление генетического алгоритма, основанное на простейшем бинарном алгоритме

Джона Холланда [3], который был использован для описания этапов работы.

Первый этап.

Генерация первой популяции. Для решения проблемы треугольников Хейльбронна требуется разместить в плоскости или пространстве точки, координаты которых вещественные, но стандартный генетический алгоритм работает с бинарными строками, из-за чего были использованы специальные формулы, с помощью которых можно было переводить вещественные числа в двоичный вид и наоборот. Также данные формулы позволяют регулировать

Генетический алгоритм



Рисунок 3 – Работа генетического алгоритма

точность вещественного числа битами, которые отводятся на каждое из них. Итак, задача алгоритма сводится к тому, чтобы сгенерировать  $n$ -ое количество бинарных строк (индивидов), заданной длины.

Второй этап.

Запись полученной популяции в массив. Данный этап является первым этапом, с которого начинается процесс эволюции популяции.

Третий этап.

Селекция. На этом этапе происходит отбор родительских индивидов по заданным заранее параметрам, эти индивиды заложат основу для индивида-ребенка, который уже пойдет в следующее поколение. Для стандартного представления генетического алгоритма был взят метод турнирной селекции.

Турнирная селекция случайным образом выбирает из популяции двух родителей, а затем сравнивает их по значению функции качества (fitness). Индивид-родитель с наибольшим качеством переходит на следующий этап.

Четвертый этап.

Кроссинговер. На этом этапе родительские особи, выбранные на этапе селекции, скрещиваются. Для стандартного представления генетического алгоритма был взят одноточечный кроссинговер.

Пятый этап.

Мутация. На этапе мутации алгоритм изменяет случайное число в бинарной строке (индивиде) с некоторой вероятностью (Рисунок 4), которая была обозначена перед запуском алгоритма. Для стандартного представления генетического алгоритма было взято три заранее подготовленные настройки значения мутации.

Сильная мутация:

$$\frac{3}{n} * 100\%$$

Средняя мутация:

$$\frac{1}{n} * 100\%$$

Слабая мутация:

$$\frac{1}{3n} * 100\%,$$

где  $n$  – длина индивида (бинарной строки).

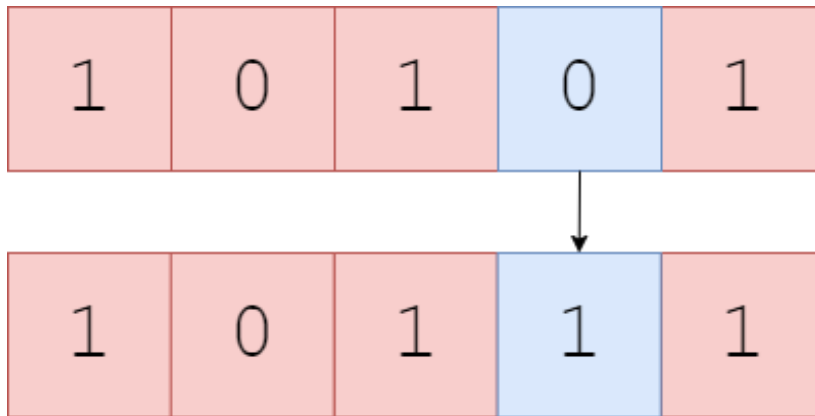


Рисунок 4 – Мутация бинарной строки

Шестой этап.

Это один из самых важных этапов, на котором производится проверка пригодности популяции. Если популяция соответствует ожидаемому результату, параметры которого задаются перед запуском алгоритма, то весь процесс эволюции со второго по шестой этап прекращается, и алгоритм выдает финальную популяцию с необходимым ответом, иначе алгоритм продолжает процесс эволюции.

Седьмой этап.

Получение итоговой популяции. Алгоритм записывает результат поиска в удобный нам вид, параметры которого задаются перед запуском. Для решения

проблемы треугольников Хейльбронна алгоритм выводил координаты точек и площадь минимального треугольника для данной конфигурации.

### 2.3 Проверка правильности работы генетического алгоритма

Стандартный генетический алгоритм (Стандартный ГА) начал свою работу, и для его проверки был написан другой алгоритм, основанный на методе Монте-Карло, который позволит нам понять эффективнее ли генетический алгоритм случайного перебора, и работает ли он корректно.

Для сравнения прилагается таблица работы двух алгоритмов для решения задачи треугольников Хейльбронна для пяти точек (Таблица 2). Используются популяции равных размеров, а также индивиды равных размеров, количество поколений в эволюции равняется 1000, вариант мутации – средняя мутация. Для статистической точности был взят усредненный результат из 100 тестов.

Настройки/Алгоритм	Монте-Карло	Стандартный ГА
1000 в популяций (16 бит)	0.0704309	<b>0.186774</b>
1000 в популяции (8 бит)	0.100148	<b>0.185828</b>
100 в популяции (16 бит)	0.60955	<b>0.185151</b>

Таблица 2 – Сравнение стандартного ГА и алгоритма Монте-Карло (жирным выделен лучший результат)

По результату, который можно увидеть на таблице, становится ясно, что алгоритм Монте-Карло гораздо менее эффективен, чем стандартный генетический алгоритм, который дал нам результаты близкие к рекомендуемым.

### 2.4 Реализованные дополнения к стандартному генетическому алгоритму

Для исследования генетических алгоритмов в решении проблемы треугольников Хейльбронна были написаны различные методы селекции и кроссинговера.

Реализованные методы селекции: турнирный, ранговый, рулеткой, случайный.

Реализованные методы кроссинговера: однотоочный, двухточечный, рулеткой, симметричный, арифметический (для вещественного алгоритма).

#### 2.4.1 Методы селекции

Турнирная селекция (Рисунок 5) случайным образом выбирает из популяции двух родителей, а затем сравнивает их по значению функции качества (fitness). Индивид-родитель с наибольшим значением качества переходит на следующий этап.

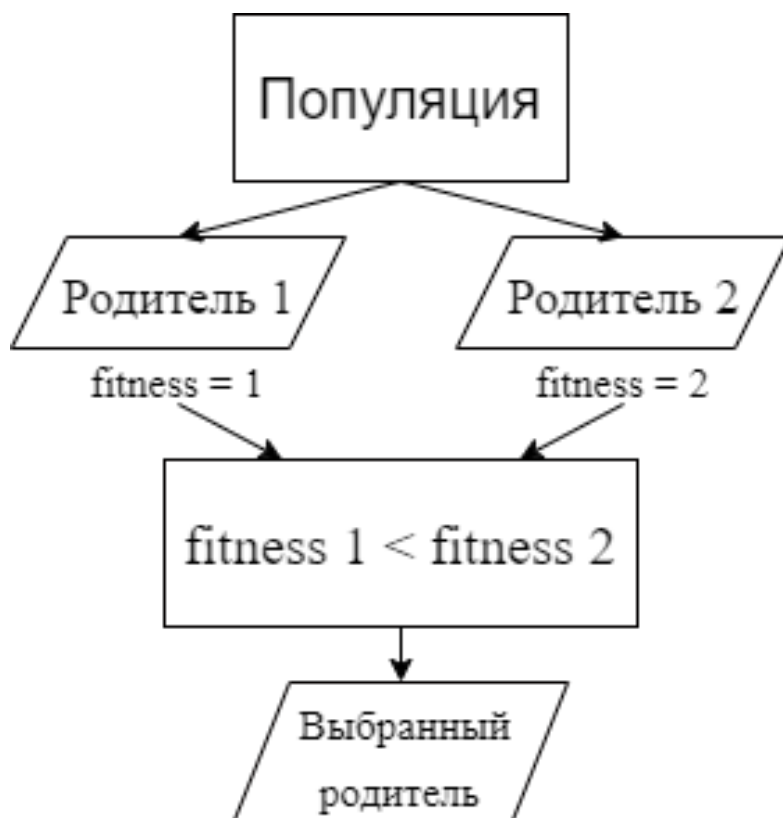


Рисунок 5 – Турнирная селекция

Ранговая селекция (Рисунок 6) сортирует всех родителей по качеству (fitness), выдавая им ранги. Родитель с наибольшим качеством получает первый ранг, а с наименьшим получает последний ранг. Чем выше ранг, тем ниже вероятность выбора.

### Ранговая селекция

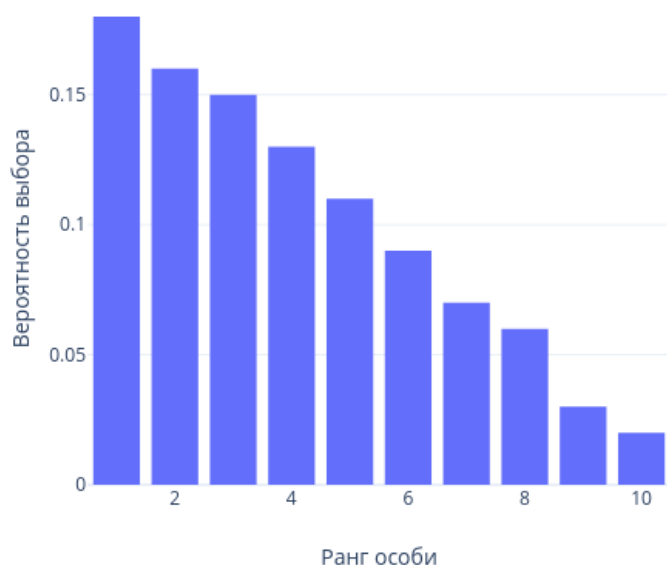


Рисунок 6 – Ранговая селекция

Селекция колесом рулетки (Рисунок 7) представляет из себя метод, в котором родительские особи выбирают пропорционально значениям их качества: каждой хромосоме сопоставлен сектор колеса рулетки, величина которого пропорционально значению качества данной хромосомы, поэтому, чем больше значение качества, тем больше сектор на колесе рулетки.

### Селекция колесом рулетки

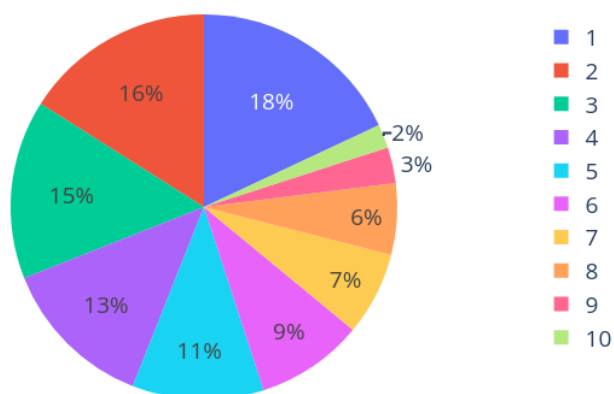


Рисунок 7 – Селекция колесом рулетки

Случайная селекция работает случайным образом: из популяции выбирается случайный родитель.

#### 2.4.2 Методы кроссинговера

Одноточечный кроссинговер (Рисунок 8) представляет из себя функцию скрещивания, в которой два родителя делятся одной точкой. Сначала правая часть первого родителя идет в индивида-ребенка, а затем левая часть второго родителя.

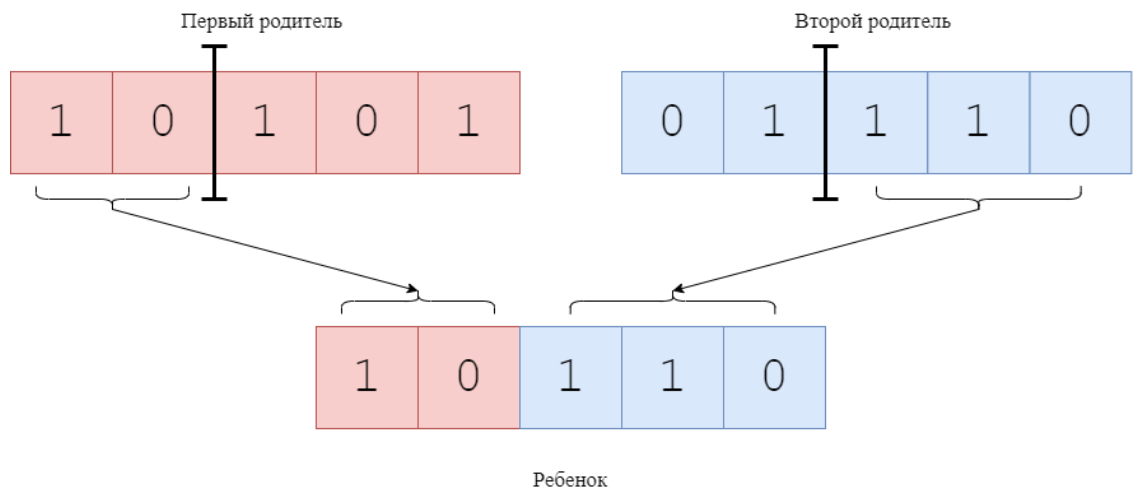


Рисунок 8 – Одноточечный кроссинговер

Двухточечный кроссинговер (Рисунок 9) является функцией скрещивания, в которой два родителя делятся двумя точками, и из полученных частей получается индивид-ребенок.

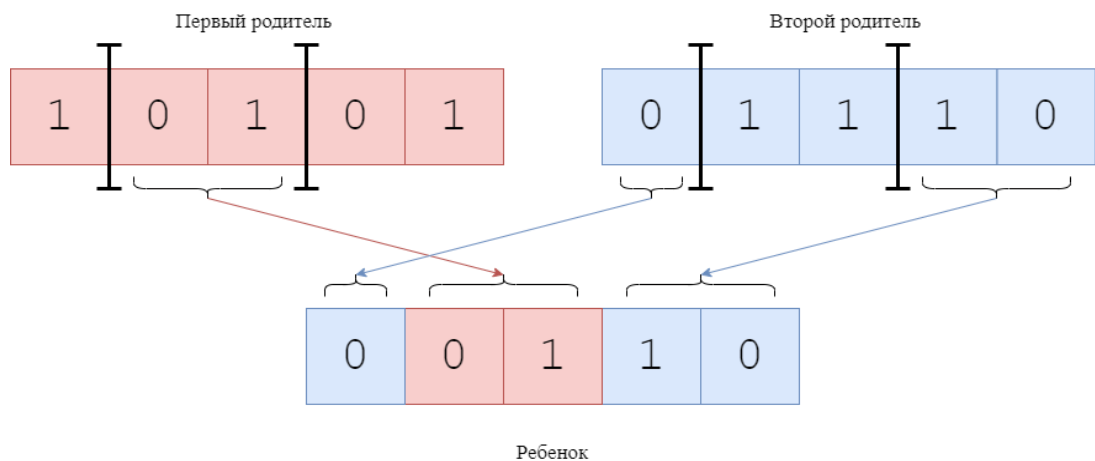


Рисунок 9 – Двухточечный кроссинговер

Кроссинговер колесом рулетки работает по тому же принципу, как селекция колесом рулетки. Выбирают двух родителей, каждой хромосоме сопоставлен сектор колеса рулетки, величина которого пропорционально качеству данной хромосомы. Далее для каждого гена хромосомы крутят колесо. Выпавший родитель кладет свой ген в индивида-ребенка.

Симметричный кроссинговер, делит хромосому первого и второго родителя наполовину, а затем получившиеся части скрещивает, получая индивида-ребенка.

#### 2.4.3 Вещественное представления хромосомы генетического алгоритма

Помимо бинарного представления хромосомы, существует вещественное представление. Данный вид генетического алгоритма имеет свои особенности. Также для него существуют свои методы кроссинговера, которые не работают в бинарном представлении (например, арифметический кроссинговер).

Арифметический кроссинговер – это наиболее используемый метод кроссинговера на данный момент. Он работает формуле (1). [7]

$$Child = \alpha * Parent1 + (1 - \alpha) * Parent2, \quad (1)$$

где  $\alpha$  – случайное вещественное число от 0 до 1;

$Child$  – хромосома ребенка;

$Parent1, Parent2$  – хромосома первого и второго родителя соответственно.

Для вещественного представления хромосомы также необходим другой метод мутации. Автором была использована формула Михалевича (2). [7]

$$v' = \begin{cases} v + \Delta(t, UB - v), & \text{если случайное число } 0. \\ v - \Delta(t, v - LB), & \text{если случайное число } 1. \end{cases} \quad (2)$$

$$\Delta(t, y) = y * (1 - r^{(\frac{1-t}{T})^b}),$$

где  $v$  – хромосома;



$UB$  – верхняя граница значения  $v$ ;

$LB$  – нижняя граница значения  $v$ ;

$t$  – номер нынешнего поколения;

$T$  – количество поколений в общем;

$r$  – случайное число от 0 до 1;

$b$  – число, отвечающее за степень неоднородности (в нашем случае  $b = 5$ , как и процент мутации).

## 3 Статистика

### 3.1 Введение

Сбор статистики необходим для получения рекомендаций по использованию генетического алгоритма при решении проблемы треугольников Хейльбронна. Прежде всего между собой сравниваются методы мутации, селекции, кроссинговера при разных обстоятельствах. Все полученные далее результаты были оценены критерием Уилкоксона.

### 3.2 Сравнение методов мутации бинарного генетического алгоритма

Для бинарной мутации были разработаны три настройки значения мутации. Каждый из них был протестирован в одинаковых условиях при разных случаях для конфигурации пяти точек. Количество поколений равно 1000. Для статистической точности были усреднены результаты 100 тестов.

Настройки ГА	Strong mutation	Medium mutation	Weak mutation
100 в популяции (8 бит)	0.125011	<b>0.185151</b>	0.12517
1000 в популяции (8 бит)	0.142524	<b>0.189677</b>	<b>0.189865</b>
100 в популяции (16 бит)	<b>0.183436</b>	0.143436	0.122084
1000 в популяции (16 бит)	<b>0.184897</b>	0.1824980	0.182873

Таблица 3 – Сравнение методов мутации (жирным выделен лучший результат)

Из результатов (Таблица 3) сравнения методов мутации бинарного алгоритма становится ясно, что сильная мутация требуется в случаях, когда каждый индивид большой длины (16 бит); среднюю мутацию лучше использовать, когда у нас большое разнообразие в популяции, содержащее в себе индивидов маленькой длины (8 бит), а слабая мутация хорошо себя показала в

случаях большого разнообразия популяции, не завися от размера индивидов. Полученные данные будут использованы в дальнейшем сборе статистики, для каждой настройки генетического алгоритма будут использованы подходящие под случай модификаторы мутации.

### 3.3 Сравнение методов селекции бинарного генетического алгоритма

Каждый метод селекции протестирован в одинаковых условиях при разных случаях для конфигурации пяти точек. Количество поколений равно 1000. Для статистической точности были усреднены результаты 100 тестов.

Настройки ГА	Турнирный метод	Ранговый метод	Метод рулеткой	Случайный метод
100 в популяции (8 бит)	0.186108	<b>0.187259</b>	0.16526	0.0731437
1000 в популяции (8 бит)	<b>0.188812</b>	0.187774	0.172252	0.0803691
100 в популяции (16 бит)	<b>0.186499</b>	<b>0.187001</b>	0.162611	0.0628605
1000 в популяции (16 бит)	<b>0.190011</b>	0.189036	0.174213	0.1080369

Таблица 4 – Сравнение методов селекции (жирным выделен лучший результат)

Полученные результаты (Таблица 4) показывают, что случайный метод селекции неэффективен, хоть и привносит большое разнообразие в популяцию. Также по данным таблицы можно понять, что турнирный метод эффективен в случаях большого разнообразия индивидов в популяции в то время, как ранговый метод селекции лучше всего подходит для случаев, когда в популяции малое количество индивидов, соответственно низкое разнообразие. Также значения турнирной и ранговой селекции незначительно отличаются в ситуации (в рамках статистической погрешности), при настройках “100 в популяции (16 бит)”.

### 3.4 Сравнение методов кроссинговера бинарного генетического алгоритма

Все методы кроссинговера протестированы в одинаковых условиях при разных случаях для конфигурации пяти точек. Количество поколений равно 1000. Для статистической точности были усреднены результаты 100 тестов.

Настройки ГА	Одноточечный кроссинговер	Двухточечный кроссинговер	Симметричный кроссинговер	Кроссинговер колесом рулетки
100 в популяции (8 бит)	0.187349	<b>0.189621</b>	0.184901	0.181799
1000 в популяции (8 бит)	0.188921	<b>1.903251</b>	0.132237	0.182326
100 в популяции (16 бит)	<b>0.186901</b>	0.182383	0.181731	0.184231
1000 в популяции (16 бит)	<b>0.190234</b>	0.188679	0.185878	0.184355

Таблица 5 – Сравнение методов кроссинговера (жирным выделен лучший результат)

Согласно результатам (Таблица 5), двухточечный кроссинговер помогает разнообразить популяцию в случаях, когда в ней малое количество индивидов, а одноточечный кроссинговер помогает не потерять качественные хромосомы при скрещивании, компенсируя малое разнообразие хромосомы большим разнообразием популяции.

### 3.5 Статистика методов селекции вещественного генетического алгоритма.

Каждый метод селекции протестирован в одинаковых условиях при разных случаях для конфигурации пяти точек. Количество поколений равно 1000. Для статистической точности были усреднены результаты 100 тестов.

Настройки ГА	Турнирный метод	Ранговый метод	Метод рулеткой	Случайный метод
100 в популяции	0.145332	0.14586	<b>0.148108</b>	0.0951437
1000 в популяции	0.171	<b>0.176179</b>	0.170141	0.07303691

Таблица 6 – Сравнение методов селекции (жирным выделен лучший результат)

Результаты таблицы (Таблица 6) показывают то, что эффективность вещественного генетического алгоритма сильно зависят от количества индивидов в популяции: чем больше индивидов, тем выше качество итоговой популяции.

### 3.6 Статистика методов кроссинговера вещественного генетического алгоритма

Методы кроссинговера протестированы в одинаковых условиях при разных случаях для конфигурации пяти точек. Количество поколений равно 1000. Для статистической точности были усреднены результаты 100 тестов.

Настрой ки ГА	Одноточечн ый кроссингов ер	Двухточечн ый кроссингов ер	Симметрич ный кроссингове р	Кроссинго вер колесом рулетки	Арифметичес кий кроссинговер
100 в популяц ии	0.14183	0.140047	0.131554	0.105582	<b>0.16324</b>
1000 в популяц ии	0.16973	0.17702	0.1683	0.135801	<b>0.18033</b>

Таблица 7 – Сравнение методов кроссинговера (жирным выделен лучший результат)

Исследуя таблицу (Таблица 7), можно сделать вывод, что арифметический кроссинговер - самый эффективный способ кроссинговера для вещественного генетического алгоритма при решении данной задачи. Также подтверждаются выводы, сделанные при сравнении методов селекции: размер популяции прямо пропорционален итоговому качеству.

### 3.7 Выводы по статистике

При решении проблемы треугольников Хейльбронна различными генетическими алгоритмами мы получили общие рекомендации по их использованию:

- Вещественные генетические алгоритмы значительно проигрывают бинарным генетическим алгоритмам по эффективности в поиске решений данной задачи.
- Арифметический кроссинговер наиболее предпочтителен для реализации вещественного генетического алгоритма.

- В маленьких популяциях лучше использовать повышенный множитель мутации, а в больших пониженный множитель.
- Индивидам большого размера рекомендуется повышенный множитель мутации, а индивидам маленького размера пониженный множитель.
- Ранговая и турнирная селекция наиболее эффективны в бинарных алгоритмах. При этом ранговая селекция подходит для маленьких популяций, а турнирная для больших.
- Одноточечный и двухточечный кроссинговер – наиболее предпочтительные методы кроссинговера для бинарного генетического алгоритма. Одноточечное скрещивание рекомендуется к использованию для кроссинговера индивидов маленького размера, а двухточечное для индивидов большого размера.

## 4 Модификации

### 4.1 Сборка оптимального генетического алгоритма

Получив необходимые данные о поведении разных вариаций генетического алгоритма, был составлен оптимальный генетический алгоритм для поиска решений проблемы треугольников Хейльбронна (Таблица 8). Как показала нам полученная статистика бинарный вещественный алгоритм наиболее эффективен в данной задаче и для наибольшей точности был отброшен вариант 8 бит на единицу хромосомы, поэтому по умолчанию стоит вариант на 16 бит. Итак, были составлены две сборки для маленькой популяции, в нашем случае 100 индивидов, и большой популяции, у нас 1000 индивидов. Для статистической точности были усреднены результаты 100 запусков по 1000 поколений.

Методы	ГА для маленькой популяции	ГА для большой популяции
Мутация	Strong mutation	Strong mutation
Селекция	Ранговая селекция	Турнирная селекция
Кроссинговер	Одноточечный кроссовер	Одноточечный кроссовер
Результаты для конфигурации пяти точек	0.186533	0.189997

Таблица 8 – Настройки оптимальных сборок генетического алгоритма

### 4.2 Модификации к оптимальным генетическим алгоритмам

Для улучшения асимптотики решений автором было принято решение добавить некоторые модификации элитарной стратегии к работе генетического алгоритма, основанные на наблюдениях, сделанных при сборе статистики.

Прежде всего стоит отметить то, что при работе генетического алгоритма могут затеряться индивиды с высоким показателем качества, так как из-за случайности они не переходят в следующее поколение, поэтому было



предпринято решение запоминать индивида с самым высоким показателем качества.

Также потерянные лучшие индивиды перестают влиять на развитие популяции, поэтому в генетический алгоритм была добавлена функция возвращения лучшего индивида в популяцию каждые десять поколений для большой популяции и каждые сто поколений для маленькой, чтобы предотвратить малое разнообразие в популяции.

Для статистической точности были усреднены результаты 100 запусков по 1000 поколений.

Методы	ГА для маленькой популяции	ГА для большой популяции
До модификаций	0.186533	0.189997
После модификаций	<b>0.1871</b>	<b>0.19052</b>

Таблица 9 – Сравнение ГА с модификациями и без модификаций (жирным выделен лучший результат)

Модификации показали свою эффективность (Таблица 9).

## 5 Симбиотические модификации

### 5.1 Введение

Симбиотические функции смешивают разные методы селекции и кроссинговера с вероятностью, заданной до запуска генетического алгоритма. Они позволяют нивелировать недостатки каждого из методов, а также привести большее разнообразие в популяцию.

### 5.2 Статистика

Для создания симбиоза было проведено исследование поведения методов селекции и кроссинговера по показателю качество (fitness) на поколение (Рисунок 10). Были усреднены показатели 10 запусков.

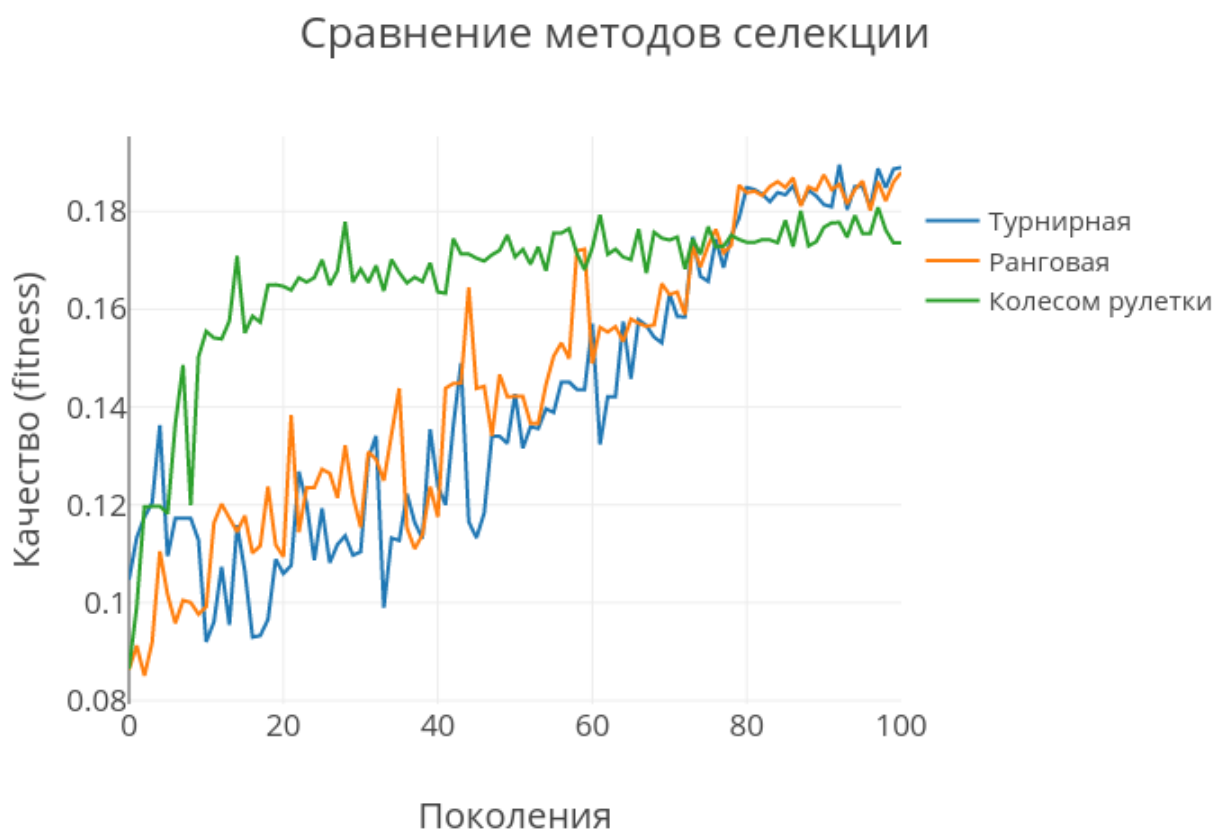


Рисунок 10 – Сравнение методов селекции (график)

## Сравнение методов кроссинговера



Рисунок 11 – Сравнение методов кроссинговера (график)

Полученные данные показали нам эффективность селекции и кроссинговера колесом рулетки в поиске локальных максимумов за минимальное количество поколений. Но, достигая локального максимума, генетический алгоритм теряет разнообразие популяции и перестает меняться в то время, как другие методы двигаются равномернее, не теряя разнообразия (кроме симметричного кроссинговера).

### 5.3 Сборка симбиоза

Исходя из знаний, полученных в предыдущем пункте, было принято решение сделать симбиоз из модифицированного генетического алгоритма и алгоритма с селекцией и кроссинговером методом колеса рулетки.

Для симбиоза была взята случайная вероятность запуска модифицированного генетического алгоритма от 70%, соответственно для селекции и кроссинговера колесом рулетки случайная вероятность до 30%.

Для статистической точности были усреднены результаты 100 запусков по 1000 поколений.

Методы	ГА для маленькой популяции	ГА для большой популяции
До модификаций	0.186533	0.189997
После модификаций	0.1871	0.19052
Симбиоз	<b>0.188823</b>	<b>0.191424</b>

Таблица 10 – Сравнение ГА, ГА с модификациями и симбиоза (жирным выделен лучший результат)

Симбиоз показал свою эффективность (Таблица 10).

## 6 Треугольники Хейльбронна

### 6.1 Треугольники Хейльбронна в плоскости

Модифицированный генетический алгоритм, показавший свою эффективность (Таблица 10), был использован в получении решения проблемы треугольников Хейльбронна для конфигурации пяти и шести точек:

- 5 точек, площадь = 0.19176 (Рисунок 12);
- 6 точек, площадь = 0.1194 (Рисунок 13).

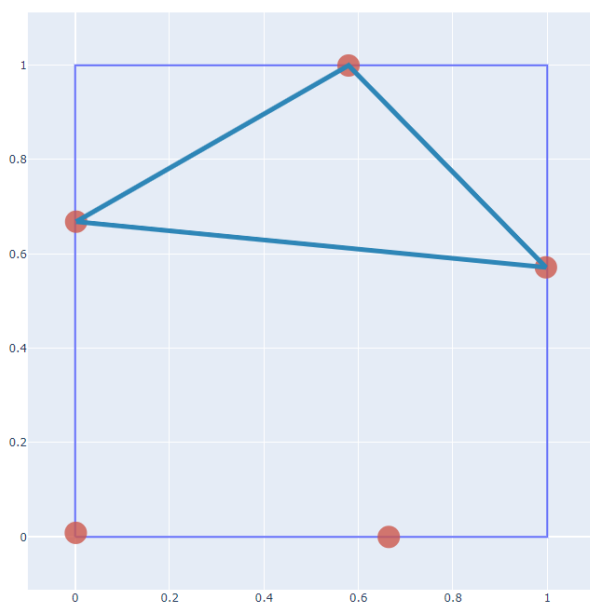


Рисунок 12 – Конфигурация 5 точек

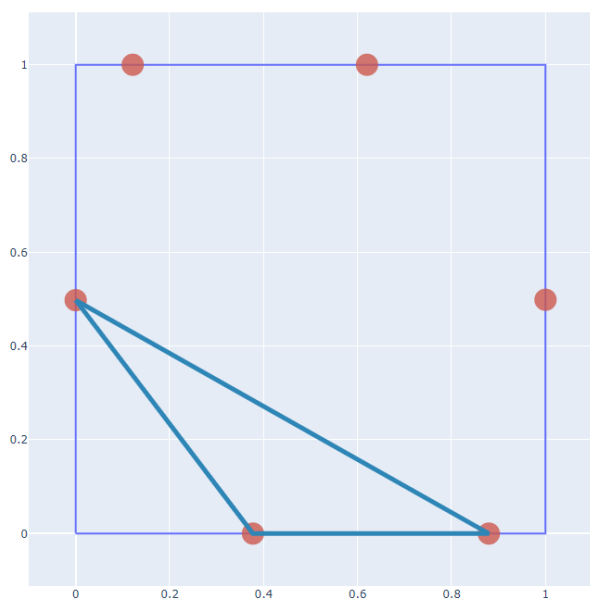


Рисунок 13 – Конфигурация 6 точек

## 6.2 Треугольники Хейльбронна в пространстве

Полученные решения для трехмерного пространства:

- 6 точек, площадь = 0.08297 (Рисунок 14);
- 7 точек, площадь = 0.05014 (Рисунок 15);
- 8 точек, площадь = 0.197192 (Рисунок 16).

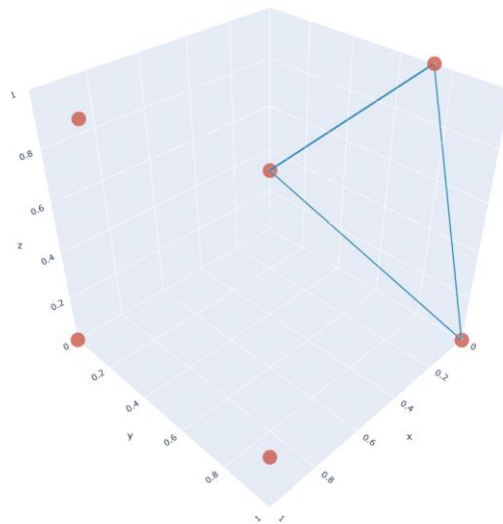


Рисунок 14 – Конфигурация 6 точек

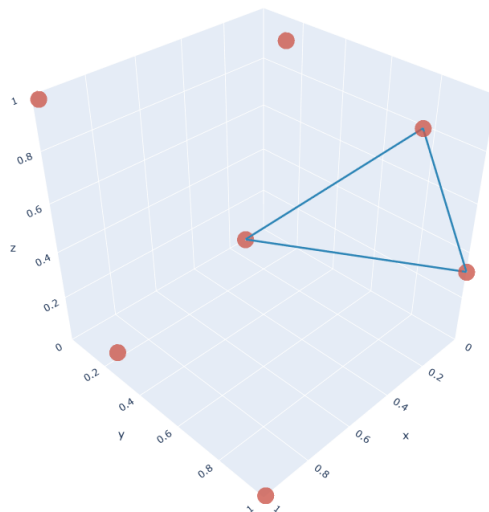


Рисунок 15 – Конфигурация 7 точек

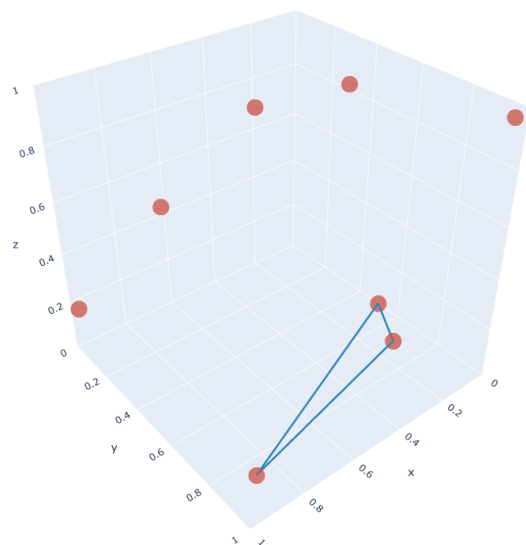


Рисунок 16 – Конфигурация 8 точек

## Выводы

В ходе работы была написана программа, решающая проблему треугольников Хейльбронна с помощью модифицированного генетического алгоритма, показавшего свое превосходство по сравнению со стандартным генетическим алгоритмом. Также был продемонстрирован сравнительный анализ алгоритма при разных настройках, а также решение задачи в трехмерном пространстве. В рамках работы были выведены следующие эмпирические правила:

- Вещественные генетические алгоритмы менее эффективны, нежели бинарные в поиске значений треугольников Хейльброна
- Одни методы повышают разнообразие популяции, другие понижают, необходимо ими чередовать, чтобы чрезмерное разнообразие не мешало поиску, а отсутствие разнообразия не загоняло в локальные минимумы или максимумы.
- Некоторые методы достигают локальных максимумов или минимумов быстрее других, но при этом имеют низкую точность поиска, так как теряют разнообразие популяции.
- Элитарная стратегия приносит хорошие результаты при умеренном использовании.
- Симбиозы эффективнее обычных генетических алгоритмов.
- Каждая задача и ситуация требует особых настроек генетического алгоритма.

Применяя эти данные, можно добиться хороших результатов при решении иных задач глобальной оптимизации.

Также существует множество модификаций генетического алгоритма, которые не были исследованы в данной работе, а также множество задач оптимизации, работающее по другим закономерностям, которые можно было бы изучить в дальнейшем.



## Список используемой литературы

1. Heilbronn Triangle Problem // mathworld.wolfram.com URL: <https://mathworld.wolfram.com/HeilbronnTriangleProblem.html> (дата обращения: 10.01.2021).
2. Cooperative and Adaptive Algorithms // uwaterloo.ca URL: <https://uwaterloo.ca/scholar/sites/ca.scholar/files/ahilal/files/lecture-6-1.pdf> (дата обращения: 22.01.2021).
3. John, H. Holland Genetic Algorithm and Adaptation. - New York: Plenum Press, 1984. - 333 с.
4. Goldbeg D., Sastry K. A Practical Schema Theorem for Genetic Algorithm Design and Tuning // GECCO'01: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. - San Francisco: Morgan Kaufmann Publishers, 2001. - С. 328–335.
5. G. Barequet, A. Shaikhet The on-line Heilbronn's triangle problem in d dimensions // Discrete & Computational Geometry. - Haifa: Springer Science+Business Media, Inc., 2007. - С. 51-60.
6. Asenath Tal Algorithms for Heilbronn's Triangle Problem : дис. ... канд. комп. наук: 1.02.00.00. - Haifa, 2009. - 80 с.
7. Michalewicz, Zbigniew Genetic Algorithms + Data Structures = Evolution Programs. - Charlotte: Springer-Verlag Berlin Heidelberg New York, 1996. - 362 с.
8. Francesc Comellas, J. Luis A. Yebra New Lower Bounds for Heilbronn Numbers // The electronic journal of combinatorics . - 2002. - № 9. - С. 1-10.
9. Heilbronn Triangle Problem // wikipedia.org URL: [https://en.wikipedia.org/wiki/Heilbronn\\_triangle\\_problem](https://en.wikipedia.org/wiki/Heilbronn_triangle_problem) (дата обращения: 10.01.2021).