МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
УРАЛЬСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
имени первого Президента России Б. Н. Ельцина

ИНСТИТУТ ЕСТЕСТВЕННЫХ НАУК И МАТЕМАТИКИ

Кафедра вычислительной математики и компьютерных наук

# МНОГОМЕРНЫЙ АЛГОРИТМ ОВЫПУКЛЕНИЯ РОЯ ТОЧЕК, НАХОДЯЩИХСЯ В НЕОБЩЕМ ПОЛОЖЕНИИ

Направление
09.04.03 «Прикладная информатика»

Допустить к защите:

Зав. кафедрой:

Магистерская
диссертация

**Корабельников**
**Алексей Алексеевич**

Нормоконтролер:

Научный руководитель:
к.ф.-м.н. доц. С. С. Кумков

Екатеринбург
2020

# РЕФЕРАТ

Корабельников Алексей Алексеевич, «МНОГОМЕРНЫЙ АЛГОРИТМ ОВЫПУКЛЕНИЯРОЯ ТОЧЕК, НАХОДЯЩИХСЯ В НЕОБЩЕМ ПОЛОЖЕНИИ», работа содержит: стр. 22, рис. 3, библ. 4 назв.

Ключевые слова: выпуклая оболочка, необщее положение точек, алгоритм.

Целью работы является разработка и реализация алгоритма построения выпуклой оболочки многомерного роя точек в случае их необщего положения. Под необщим положением точек подразумевается ситуация, когда в гиперплоскости $R^n$ измерения лежит больше чем $n + 1$ точка. Наиболее перспективный с целью переноса в более высокую размерность оказался метод заворачивания подарка, поскольку он минимально использует специфику плоскости. В результате работы была разработана реализация алгоритма заворачивания подарка для работы в многомерном пространстве при необщем положении точек.

# СОДЕРЖАНИЕ

# Введение

Одна из фундаментальных проблем вычислительной геометрии является построение выпуклой оболочки конечного набора точек. На текущий момент, существует множество алгоритмов, решающих эту проблему. Эти алгоритмы широко применяются для решения задач в геометрическом моделировании, распознавании образов, экономике, статистики и в многом другом.

При работе алгоритмов может возникнуть проблема необщего положения точек. Если в гиперплоскости $R^n$ измерения лежит больше чем $n+1$ точка, возникает проблема построения грани в виде симплекса, в результате чего получается некорректно построенная выпуклая оболочка.

Для двухмерного и трехмерного случая уже существуют способы решения этой проблемы. Для многомерного случая однозначного способа решения пока нет и требуется тщательная проработка реализации метода.

Из существующих алгоритмов только несколько можно расширить для обработки точек высокой размерности. Самый перспективный с целью переноса в более высокую размерность является метод заворачивания подарка, поскольку он минимально использует специфику плоскости.

Поэтому в рамках выпускной квалификационной работы была поставлена цель разработать алгоритм построения выпуклой оболочки конечного набора точек $R^n$ размерности, находящихся в необщем положении.

# 1 Основная часть

## 1.1 Обзор литературы

Здесь будет рассмотрены построение выпуклой оболочки в двухмерном и трехмерном пространстве, при помощи алгоритмов заворачивания подарка. Для этого вначале определим, что такое выпуклая оболочка.

Пусть $P = \{p_1, p_2 .....p_m\}$ - конечное множество точек в $n$-мерном евклидово пространстве. Пересечение всех выпуклых подмножеств $R^3$, содержащих $P$, называется выпуклой оболочкой $P$ и обозначается через $CH(P)$.[3]

Предполагается, что $P$ содержит по крайней мере три точки, и они не являются коллинеарными. $P$ называется копланарным, если все точки в $P$ находятся в общей плоскости, а некопланарным в противном случае.

Если $P$ копланарно, $CH(P)$ - это выпуклый многоугольник $n-1$-мерного пространства, если $P$ некопланарно, $CH(P)$ представляет собой выпуклый многогранник $n$-мерного пространства.

### 1.1.1 Алгоритм Джарвиса на плоскости

Пусть дано множество точек $P = \{p_1, p_2, \ldots, p_n\}$. Для построения выпуклой оболочки необходимо задать начальную точку $p_1$, которая точно является вершиной выпуклой оболочки. Для этого берем самую левую нижнюю точку. Вторую точку $p_2$ берем такую, которая образует наибольший угол между векторами $\overline{n_1} = (1, 0)$, $\overline{p_1 p_2}$.

После этого для каждой точки $p_i \, (2 < i \leq |P|)$ из оставшихся точек ищется такая точка $p_{i+1}$, в которой будет образовываться наибольший угол между векторами $\overline{p_i p_{i-1}}$ и $\overline{p_i p_{i+1}}$. Пример нахождения следующей вершины приведен на рисунке 1.

Нахождение вершин выпуклой оболочки продолжается до тех пор, пока $p_{i+1} \neq p_1$. В тот момент, когда следующая точка в выпуклой оболочке совпала с первой, алгоритм останавливается - выпуклая оболочка построена.

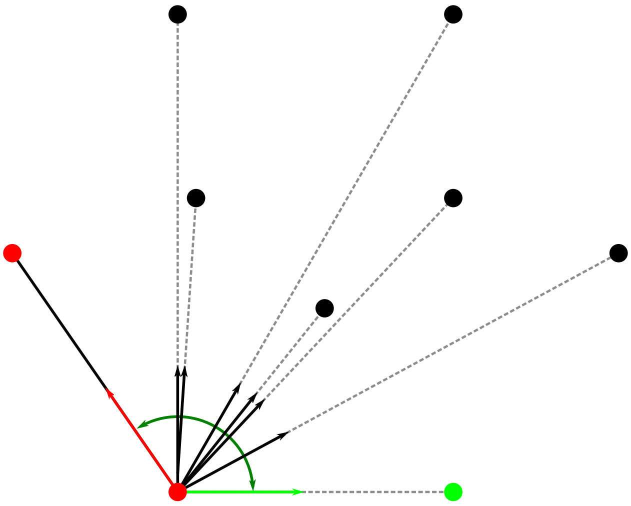

Рис. 1: Алгоритм Джарвиса. Определение следующей вершины

### 1.1.2 Алгоритм Джарвиса на пространстве

Алгоритм является обобщением алгоритма заворачивание подарка на плоскости. Вместо прямой поворачивают плоскостью. Алгоритм описан в листинге 1.

**Листинг 1:** Алгоритм заворачивания подарка в $R^3$ - CH(P)

**Входные данные:** Множество точек $P$

$P = \{p_i | p_i = (x_1, x_2, \ldots, x_n) \in R^3, i = [1 \ldots n]\}$

**Результат:** Выпуклая оболочка множества точек $CH(P)$, заданная набором граней

Инициализировать пустой массив $C$

Найти начальную грань $f_0$ выпуклой оболочки

Добавить грань $f_0$ в $C$

Добавить грань $f_0$ в очередь $Q$

**до тех пор, пока** $Q \neq \varnothing$ **выполнять**

    Выбрать и удалить грань $f$ из $Q$

    Добавить все ориентированные ребра $f$ в $A$

    **для каждого** *ребра* $e = p_i p_j \in A$ **выполнять**

        Найти грань $f'$, состоящую из точек $\{p_j, p_i, p_l\}$ и образующий максимальный угол с гранью $f$

        **если** $f' \notin C$ **тогда**

            Добавить в в очередь $Q$ и массив $C$

**возвратить** $C$

### 1.1.3 Необщее положение точек

При работе алгоритмов может возникнуть проблема необщего положения точек. Если в гиперплоскости $R^n$ измерения лежит больше чем $n+1$ точка, возникает проблема построения грани в виде симплекса, в результате чего получается некорректно построенная выпуклая оболочка.

Метод Джарвиса на плоскости достаточно легко можно расширить для работы при необщем положении. Когда на прямой лежит больше чем 2 точки, берем самую дальнюю.

Но в трехмерном измерении и выше возникают более серьезные проблемы, не позволяющие корректно построить выпуклую оболочку.

## 1.2 Постановка задачи работы

Дано конечное множество точек $P = \{p_1,\ p_2, \ldots,\ p_m\}$ точек $p_i = (x_1^i,\ x_2^i\ , \ldots,\ x_n^i)$ $R^n$-мерного евклидова пространства.

Точки могут находится в необщем положение, то есть в любой гиперплоскости $n$-мерного евклидова пространства может лежать больше чем $n + 1$ точка. Но основан на В результате чего выпуклая оболочка может иметь несимплициальные грани.

Требуется построить выпуклую оболочку, т.е. найти наименьший выпуклый многомерный многогранник, содержащий все точки множества P.

## 1.3 Реализация алгоритма

В результате практической работы был разработан алгоритм построения выпуклой оболочки. (см. листинг 2) За основу алгоритма были взяты принципы работы алгоритма заворачивания подарка, поскольку он минимально использует специфику плоскости.

Рассмотрим работу алгоритма 2. В строке 1 происходит проверка, что точки являются двухмерными. Если это так, то вызывается метод построения выпуклой оболочки в двухмерном пространстве и алгоритм заканчивает работу. Алгоритм построения выпуклой оболочки на плоскости описан в разделе 1.1.1.

В строке 6 производится поиск начальной гиперграни. Алгоритм описан в листинге 3.

Далее просходит заполнение очереди $S$ гиперребрами начальной гиперграни. Очередь $S$ хранит гиперребра, к которым еще не было найдено соседней гиперграни.

В множестве "H"(см. строку 5) хранятся список гиперребер, у которых уже были найдены соседние гиперграни. Это необходимо для заполнения очереди $S$ свободными гиперребрами из новонайденных гиперграней. Проверка и добавление гиперребер найденной гиперграни проиходит в строке 21.

В строке 13 проверяется, что являются ли найденные точки $P_g$ гиперплоскости симплексом. Если является, то вызывается отдельный метод построения симплицированной гиперграни. Иначе происходит перенос точек $P_g$ в базис гиперплоскости и рекурсивный вызов метода для обработки множества точек $P_g$ .

**Листинг 2:** Алгоритм построения выпуклой оболочки - $CH(P)$

**Входные данные:** Множество точек $P$

$P = \{p_i | p_i = (x_1, x_2, \ldots, x_n) \in R^n, i = [1 \ldots m]\}$

**Результат:** Выпуклая оболочка множества точек $CH(P)$, заданная набором граней.

**если** $P \in R^2$ **тогда**
- Найти выпуклую оболочку $CH_{2d}$ на плоскости при помощи метода Джарвиса
- **возвратить** $CH_{2d}$

Инициализировать пустую выпуклую оболочку $CH$

Инициализировать пустое множество использованных ребер $H$

Найти исходную грань $F_0$ выпуклой оболочки

Добавить ребра $F_0$ в очередь $S$

**до тех пор, пока** $S \neq \varnothing$ **выполнять**
- Выбрать и удалить ребро $e$ из $S$
- Добавить $e$ в $H$
- Найти гиперплоскость $g$, содержащую $CH(e \cup p)$, где $p \in P$, и все точки $p_i \in P$ принадлежат нижнему полупространству плоскости $\pi$
- Найти множество точек $P_g = \{p_i | p_i \in P, p_i \in g, i = [1..m_1]\}$
- **если** $m_1 = n$ **тогда**
  - Найти симплицированную грань $F$
- **иначе**
  - Найти координаты $P_g$ в базисе гиперплоскости $g$
  - $F = \mathrm{CH}(P_g)$
- **если** $P = P_g$ **тогда**
  - **возвратить** $F$
- Добавить грань F в CH
- Добавить ребра $E = \{e_i | e_i \in F, e_i \notin H, i = [1 \ldots m_2]\}$ в $S$

**возвратить** CH

**Листинг 3:** Алгоритм поиска первой грани - FindFirstFace(P)

**Входные данные:** Множество точек $P$

$P = \{p_i | p_i = (x_1, x_2, \dots, x_n) \in R^n, i = [1 \dots m]\}$

**Результат:** Выпуклая оболочка множества точек $CH(P)$, заданная набором граней

```
Инициализировать пустое множество точек H
Найти минимальную точку p_min множества P
Добавить p_min в H
Найти плоскость hp проходящую через точку p_min с нормалью
  n̅ = (1, 0, 0, ..., 0, 0)
цикл n − 1 раз выполнять
    Поворотом плоскости вокруг вектора n̅ найти гиперплоскость
      hp_next проходящую через точки H ∪ p_next, где p_next ∈ P и
      дающая максимум скалярного произведения с плоскостью hp
    Добавить p_next в H
    hp = hp_next
Найти координаты точекы H в базисе гиперплоскости hp
возвратить CH(H)
```

# Заключение

В данной работе был разработан алгоритм решение задачи построения выпуклой оболочки конечного набора точек $R^n$ размерности, находящихся в необщем положении.

Алгоритм не является окончательным и требует дальнейшей доработки.

Оставшиеся задачи и алгоритм будут доработаны в будущей выпускной работе.

# Список литературы

[1] Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. под ред. Банковского Ю. М. — М.: Мир, 1989. — 478 с.

[2] Ивановский С. А., Преображенский А. С., Симончик С. К. Алгоритмы вычислительной геометрии. Выпуклые оболочки в трехмерном пространстве // КИО. 2007.

[3] Kokichi Sugihara, Robust gift wrapping for the three-dimensional convex hull // Journal of Computer and System Sciences, 1994. 391-407 с.,