Алгоритм овыпукления в $\mathbb{R}^d$

1 Обозначения и определения

Евклидово пространство $\mathbb{R}^d$, $d > 1$. Его элементы обозначим $\vec{x} = (x_i)_{i=1}^d$.

Ноль-вектор будем обозначать $\vec{0}$.

Аффинное пространство $\mathbb{A}^d$, $d > 1$. Его элементы обозначим $x = [x_i]_{i=1}^d$.

Элементу $x \in \mathbb{A}^d$ сопоставляется элемент $\vec{x} \in \mathbb{R}^d$, который есть радиус-вектор точки x.

Определены операции умножения на скаляр, сложения и вычитания. Над элементами $\mathbb{R}^d$ они определяются классическим образом как операции линейного пространства. Для аффинного пространства $\mathbb{A}^d$ определения следующие:

- $\forall \alpha \in \mathbb{R}\ \forall x = [x_i] \in \mathbb{A}^d\ \alpha \cdot x = \alpha x = [\alpha \cdot x_i] \in \mathbb{A}^d$;
- $\forall a \in \mathbb{A}^d\ \forall \vec{x} \in \mathbb{R}^d\ a + \vec{x} = [a_i + x_i] \in \mathbb{A}^d$;
- $\forall a, b \in \mathbb{A}^d\ a - b = (a_i - b_i) \in \mathbb{R}^d$.

Линейная независимость векторов из $\mathbb{R}^d$ понимается в классическом смысле линейной независимости в линейном пространстве.

Линейная независимость набора точек $\{a_k\}_{k=0}^m \subset \mathbb{A}^d$, $m \leqslant d$, понимается как линейная независимость набора векторов $\{a_k - a_0\}_{k=1}^m \subset \mathbb{R}^d$.

k-мерным симплексом (k-симплексом) в пространстве $\mathbb{A}^d$, $k \leqslant d$, назовём линейно независимый набор $k+1$ точек.

Базис линейного пространства $\mathbb{R}^d$ понимается в классическом смысле. Аффинный базис аффинного пространства $\mathbb{A}^d$ может представляться в двух эквивалентных формах: либо как d-симплекс этого пространства, либо как пару $\left(o, \{\vec{x}_i\}_{i=1}^d\right)$, $o \in \mathbb{A}^d$, $\vec{x}_i \in \mathbb{R}^d$, такую что набор точек $\{o, o + \vec{x}_i\}_{i=1}^d$ линейно независим.

Назовём $(d-1)$-грань d-многогранника *симплициальной*, если она имеет ровно d вершин (является $(d-1)$-симплексом).

Под словом «плоскость в пространстве $\mathbb{R}^d$» будем понимать гиперплоскость размерности $\mathbb{R}^{d-1}$.

Вектор $\vec{e}_i = (\delta_{i,k})_{k=1}^d$ – i-й вектор ортонормированного базиса $\mathbb{R}^d$. Здесь $\delta_{i,k}$ – символ Кро́некера:

$$\delta_{i,k} = \begin{cases} 1, & i = k, \\ 0, & i \neq k. \end{cases}$$

Обозначим через $\mathcal{N}(\vec{x})$ операцию нормирования вектора: $\mathcal{N}(\vec{x}) = \vec{x}/\|\vec{x}\|$. Считаем, что $\mathcal{N}(\vec{0}) = \vec{0}$.

Обозначим через $\mathcal{ON}(\vec{v}, \mathcal{B})$ операцию ортонормирования вектора $\vec{v}$ на фоне ортонормированного набора векторов $\mathcal{B}$. Выполняется с использованием алгоритма Грамма-Шмидта. Считаем, что $\mathcal{ON}(\vec{a}, \varnothing) = \mathcal{N}(\vec{a})$.

2 Построение начальной грани $\mathcal{P}$

Пусть $\mathcal{S} \subset \mathbb{R}^d$ – рой d-мерных точек в d-мерном пространстве. Построение начальной грани состоит из двух шагов. Первый – поиск аффинной плоскости, содержащей какую-либо грань выпуклой оболочки $\mathcal{P}$. Второй – поиск вершин этой грани. Построение аффинной плоскости заключается в последовательном повороте некоторой начальной плоскости, проходящей через одну точку роя. Каждый поворот аффинной плоскости заключается в подмене одного вектора из её базиса так, чтобы она проходила через ещё хотя бы одну точку роя. Когда очередная плоскость содержит d линейно независимых точек, искомая плоскость построена.

Временные векторы базиса $\mathbf{T}$ – векторы из этого множества могут быть заменены.

Финальные векторы базиса $\mathbf{F}$ – векторы базиса, которые далее не будут заменяться и будут входить в базис искомой плоскости.

2.1 Построение начальной плоскости $\mathcal{L}$ по мотивам Сварта

Представленная процедура опирается на алгоритм из статьи[1].

Вход: $\mathcal{S} \subset \mathbb{R}^d$ – рой точек в d-мерном пространстве.

Выход: Начальная плоскость $\mathcal{L} \subset \mathbb{R}^d$, заданная аффинным базисом $(o, \mathcal{L}_\mathcal{B})$, где размерность линейного базиса равна $d-1$. Внешняя нормаль $\vec{n}$ к этой плоскости.

1. Выберем точку $o \in \mathcal{S}$, минимальную в лексикографическом порядке (она гарантировано будет вершиной $\mathcal{P}$);

2. Проведём через $o = (o_i)_{i=1}^d$ плоскость $\mathcal{L}$ перпендикулярно первому базисному вектору пространства. (Все точки роя $\mathcal{S}$ гарантированно лежат не левее этой плоскости.) Положим $\mathbf{F}_0 \leftarrow o$.

3. Положим нормаль к плоскости $\vec{n}_0 \leftarrow -\vec{e}_1$.

4. Пока $|\mathbf{F}_k| < d-1$, повторяем:

   (a) Имеем $\vec{n}_k$ — нормаль к текущей плоскости, $\mathbf{F}_k$ — накопленная часть базиса плоскости такая, что $(o, \mathbf{F}_k)$ — аффинный базис тех точек, которые уже включены в плоскость.

   (b) Вычислим вектор $\vec{e}_*$, перпендикулярный оси вращения. Перебираем все орты $\vec{e}_i$ и для очередного вектора $\vec{e}_i$ вычисляем вектор: $\vec{v} = \mathcal{ON}(\vec{e}_i, \mathbf{F}_k \cup \{\vec{n}_k\})$. В качестве искомого вектора $\vec{e}_*$ положим первый ненулевой вектор $\vec{e}_i$.

   (c) Пусть $s \in \mathcal{S}$. Рассмотрим вектор $\vec{u} = \langle s-o, \vec{v}\rangle \vec{v} + \langle s-o, \vec{n}_k\rangle \vec{n}_k$ — вектор, перпендикулярный оси вращения и лежащий в новой плоскости, содержащей точку s. Если $\vec{u} = \vec{0}$, то есть точка s лежит в оси вращения, то переходим на шаг 4c.

   (d) Иначе найдём вычислим угол между $\vec{v}$ и $\vec{u}$. Если он больше текущего максимального угла, то запоминаем его, точку s и вектор $\mathcal{N}(u)$. Переходим на шаг 4c.

   (e) Если точка s_* не нашлась, это означает, что весь рой $\mathcal{S}$ лежит в аффинном подпространстве размерности меньше $d-1$. В этом случае или алгоритм прекращает работу, если целью было найти выпуклую оболочку полной размерности, или переходит к построению выпуклой оболочки роя $\mathcal{S}$ в найденном аффинном подпространстве с базисом $(o, \mathbf{F}_k)$.

   (f) Если таких экстремальных точек несколько, то можно выбрать любую. Расширим финальный базис: $\mathbf{F}_{k+1} \leftarrow \mathbf{F}_k \cup \big\{\mathcal{ON}(s_* - o, \mathbf{F}_k)\big\}$.

   (g) Вычислим нормаль $\vec{n}_{k+1}$ новой плоскости: $\vec{n}_{k+1} = \langle \vec{u}^*, \vec{n}_k\rangle \vec{v} - \langle \vec{u}^*, \vec{v}\rangle \vec{n}_k$. Выполним переориентацию нормали (см. 4.1). После нужно проверить, что есть хотя бы одна точка роя $\mathcal{S}$ вне плоскости $\mathcal{L}$. Если такой точки не нашлось, значит все точки лежат в аффинном подпространстве с базисом $(o, \mathbf{F}_{k+1})$. Аналогично пункту 4e либо останавливаем алгоритм, либо переходим к построению выпуклой оболочки в этом аффинном подпространстве.

[1] Swart, Garret. "Finding the Convex Hull Facet by Facet." J. Algorithms 6(1) (1985): 17–48.

5. Положим аффинный базис начальной плоскости (o, F_k), $\vec{n} = \vec{n}_k$.

6. Конец процедуры.

Заметим, что в пункте 4g происходит накопление ошибки. Однако при вычислении начальной плоскости пересчётов нормали немного, максимум d, поэтому на результат ошибка не успеет оказать влияния. В качестве альтернативы можно воспользоваться следующей процедурой:

1. Перебираем орты e_i вычисляя $n_{k+1} = \mathcal{ON}(\vec{e}_i, \mathbf{F}_k)$. Выполняем пока $n_{k+1} = 0$.

2.2 Построение начальной плоскости $\mathcal{L}$ нашим методом.

Вход: $\mathcal{S} \subset \mathbb{R}^d$ – рой точек в d-мерном пространстве.

Выход: Начальная плоскость $\mathcal{L} \subset \mathbb{R}^d$, заданная аффинным базисом $(o, \mathcal{B})$, где размерность линейного базиса равна $d-1$. Внешняя нормаль $\vec{n}$ к данной плоскости.

1. Выберем точку $o \in \mathcal{S}$, минимальную в лексикографическом порядке (она гарантировано будет вершиной $\mathcal{P}$);

2. Проведём через $o = (o_i)_{i=1}^d$ плоскость $\mathcal{L}$ перпендикулярно первому базисному вектору пространства. (Все точки роя $\mathcal{S}$ гарантированно лежат не левее этой плоскости.) Положим $\mathbf{F}_0 = \{o\}$, $\mathbf{T}_0 = \{\vec{e}_i\}_{i=2}^d$.

3. Обозначим $\mathcal{V}$ – множество просмотренных вершин плоскости $\mathcal{L}$. Положим $\mathcal{V} = \{o\}$.

4. Пока $\mathbf{T}_k \neq \varnothing$, повторяем:

   (a) Возьмём произвольный $\vec{t} \in \mathbf{T}_k$. Удалим вектор $\vec{t}$ из $\mathbf{T}_k$: $\mathbf{T}_k \leftarrow \mathbf{T}_k \setminus \{\vec{t}\}$. Будем вращать вокруг ребра, аффинный базис которого есть (o, E), где $E = \mathbf{F}_k \cup \mathbf{T}_k$.

      i. Будем искать точку s_* такую, что на ней достигается максимум угла между t и $s - o$.

      ii. Берём очередную точку $s \in \mathcal{S}$, $s \notin \mathcal{V}$. Пусть $\vec{u}_F = \mathcal{ON}(s - o, \mathbf{F}_k)$. Если $\vec{u}_F = \vec{0}$, то есть набор $\mathbf{F}_k \cup \{s - o\}$ линейно зависим, то добавляем точку s в множество $\mathcal{V}$ и переходим на шаг 4(a)ii.

      iii. Иначе вычислим вектор $\vec{u} = \mathcal{ON}(\vec{u}_F, \mathbf{T}_k)$. Если $\vec{u} = \vec{0}$, то есть набор векторов $\mathbf{T}_k \cup \{\vec{u}_F\}$ линейно зависим, то переходим на шаг 4(a)ii. (В этом случае мы не добавляем точку во множество просмотренных вершин, так как она может попасть в выпуклую оболочку роя).

      iv. Иначе вычислим угол между $\vec{u}$ и $\vec{t}$. Если он больше текущего максимума, то запомнить текущий угол и данную точку s. Угол между двумя векторами может быть вычислен как арккосинус от скалярного произведения векторов делённых на их длину. Переходим на шаг 4(a)ii.

   (b) Если точка s_* не нашлась, это означает, что весь рой $\mathcal{S}$ лежит в аффинном подпространстве размерности меньше $d-1$. В этом случае или алгоритм прекращает работу, если целью было найти выпуклую оболочку полной размерности, или переходит к построению выпуклой оболочки роя $\mathcal{S}$ в найденном аффинном подпространстве с базисом $(o, \mathbf{F}_k)$.

   (c) Если таких экстремальных точек несколько, то можно выбрать любую. Пусть $\vec{v} = \mathcal{ON}(s_* - o, \mathbf{F}_k)$. Добавим вектор $\vec{v}$ в финальный базис: $\mathbf{F}_{k+1} \leftarrow \mathbf{F}_k \cup \{\vec{v}\}$. Добавим точку s_* в множество $\mathcal{V}$.

(d) Пересчитаем $\mathbf{T}_k$ в $\mathbf{T}_{k+1}$ на фоне $\mathbf{F}_{k+1}$.

i. $\mathbf{T}' \leftarrow \varnothing$.

ii. Для всех векторов $\vec{t} \in \mathbf{T}_k$ ортогонализируем его на фоне текущего финального базиса, после чего ортогонализируем на фоне текущего накопленного $\mathbf{T}'$ и добавляем получившийся вектор в $\mathbf{T}'$:

iii. $\mathbf{T}' \leftarrow \mathcal{ON}(\vec{t}, \mathbf{F} \cup \mathbf{T}') = \mathcal{ON}(\mathcal{ON}(\vec{t}, \mathbf{F}), \mathbf{T}')$.

iv. Положим $\mathbf{T}_{k+1} \leftarrow \mathbf{T}'$.

(e) Возвращаемся в начало цикла.

5. Вычислим внешнюю нормаль $\vec{n}_*$ к начальной плоскости (см. 4.2).

6. Положим $\vec{n} \leftarrow \vec{n}_*$, аффинный базис начальной плоскости $(o, \mathbf{F})$.

7. Конец процедуры.

2.3 Построение грани

Вход: Рой точек $\mathcal{S} \subset \mathbb{R}^d$ и плоскость $\mathcal{L} \subset \mathbb{R}^d$, то есть аффинный базис $(o, \mathcal{B})$, размерность базиса $k = d - 1$. Возможна передача $(d-2)$-ребра $\mathcal{E}$, являющегося начальной гранью для процедуры овыпукления в $(d-1)$-пространстве плоскости $\mathcal{L}$.

Выход: $(d-1)$-грань $\mathcal{F} \subset \mathbb{R}^d$, лежащая в плоскости $\mathcal{L}$.

1. Пусть $\mathcal{S}_{\mathcal{L}}$ — множество точек $\mathcal{S} \subset \mathbb{R}^d$, лежащих в плоскости $\mathcal{L}$.

2. Если $|\mathcal{S}_{\mathcal{L}}| = d$, то полученная грань симплициальна, дальнейшая обработка не требуется. Искомая грань — симплекс, построенный на d данных точках.

3. Иначе проецируем $\mathcal{S}_{\mathcal{L}}$ на $\mathcal{L}$ используя аффинный базис (предварительно запоминая из какой точки $s \in \mathcal{S}$ получилась очередная точка $s' \in \mathcal{L}$) и рекурсивно строим выпуклую оболочку в аффинном подпространстве плоскости $\mathcal{L}$, если передано ребро $\mathcal{E}$, то проецируем его на $\mathcal{L}$, в качестве нормали к нему устанавливаем $\vec{e}_k$ (вектор размерности $d-1$, где на последней позиции стоит единица) и передаём его в процедуру овыпукления в подпространстве (см. **??**).

4. Результатом овыпукления в подпространстве плоскости будет $(d-1)$-многогранник выраженный в терминах $(d-1)$-пространства, поэтому «поднимаем» его в d-мерное исходное пространство: очередную вершину s' подменяем исходной для неё точкой s. Производим аналогичную операцию для всех дочерних объектов данного многогранника.

5. На данном этапе можно убрать некоторые вершины из роя $\mathcal{S}$. В результате овыпукления в подпространстве получился набор V вершин выпуклого $(d-1)$-гранника. Значит из $\mathcal{S}$ можно выкинуть все точки содержащиеся в разности $\mathcal{S}_{\mathcal{L}}$ и V. Данные точки точно не будут состоять в выпуклой оболочке всего роя $\mathcal{S}$.

6. Конец процедуры.

3 Процесс заворачивания

Можно рассмотреть граф граней искомой выпуклой оболочки $\mathcal{P}$. Вершины графа сопоставляются с гранями $\mathcal{P}$. Две вершины являются соседними, если соответствующие им $(d-1)$-грани имеют общее $(d-2)$-ребро.

Если требуется построить выпуклую оболочку в двумерном пространстве, то запускается какой-либо плоский алгоритм построения выпуклой оболочки. Иначе запускается следующий многомерный алгоритм, вообще говоря, рекурсивный по размерности овыпукляемых роёв.

В начале процесса заворачивания с помощью алгоритма из предыдущего раздела нам становится известна какая-то вершина этого графа. Также нам известны рёбра графа, выходящие из этой вершины, так как нам известны рёбра этой начальной грани.

В таком рассмотрении построение выпуклой оболочки соответствует обходу всех её граней, то есть обходу графа граней $\mathcal{P}$. Такой обход графа может быть осуществлён каким-либо поисковым алгоритмом. Наиболее компактную реализацию имеет алгоритм поиска в глубину. Эта реализация является рекурсивной.

Напомним, что один рекурсивный шаг поиска в глубину состоит в переборе всех не посещённых соседей текущей вершины и переходов в них с продолжением поиска оттуда. В геометрических терминах перебор соседей и переход в них соответствует перебору рёбер текущей грани и построению грани, соседней текущей через очередное рассматриваемое ребро. Такое построение осуществляется поворотом плоскости текущей грани вокруг рассматриваемого ребра до касания какой-либо точки роя $\mathcal{S}$, не лежащей на рассматриваемом ребре.

3.1 Заворачивание подарка

Вход: $\mathcal{S} \subset \mathbb{R}^d$, возможна передача информации о начальной $(d-1)$-грани $\mathcal{F} \subset \mathbb{R}^d$, она должна содержать информацию о внешней нормали.

Выход: d-многогранник $\mathcal{P} \subset \mathbb{R}^d$.

1. Если $d = 2$, то запускается алгоритм двумерного овыпуклења, например ArcHull или GrahamScan. Тогда $\mathcal{P}$ создаётся на основе их результата.
2. Если не передали начальную грань:
   (a) Выполняем построение начальной плоскости см. 2.1 или 2.2.
   (b) Строим начальную грань $\mathcal{F}$ см. 2.3.
3. Проходимся по всем $(d-2)$-рёбрам начальной грани и увеличиваем счётчик смежных граней на один. (Изначально он равняется нулю.)
4. Пусть Q – очередь. В ней будут лежать грани с которых можно перекатиться на соседнюю грань. Добавим в неё начальную грань. Пока $Q \neq \varnothing$:
   (a) Пусть $\mathcal{F}$ это очередной взятый элемент из Q.
   (b) Пусть $\mathcal{E}$ это ребро $\mathcal{F}$ у которого счётчик смежных граней равняется одному (то есть через него нужно перекатиться).
   (c) Перекатываемся (3.2) через ребро $\mathcal{E}$ с грани $\mathcal{F}$. Получили новую грань $\mathcal{F}'$
   (d) У всех ребёр новой грани увеличиваем счётчик смежных граней на один. Если хотя бы у одного ребра значение счётчика оказалось равным единице, то добавляем новую грань в очередь: $Q \leftarrow \mathcal{F}'$.

5. «Собираем» из полученных $(d-1)$-граней d-многогранник $\mathcal{P}$.

6. Конец процедуры.

3.2 Процедура перекатывания через ребро

Вход: $\mathcal{S} \subset \mathbb{R}^d$, текущая $(d-1)$-грань $\mathcal{F}$ и $(d-2)$-ребро $\mathcal{E}$ текущей грани, через которое происходит перекатывание, $(o, \mathcal{B}_{\mathcal{F}})$ — аффинный базис плоскости содержащей грань $\mathcal{F}$. Вектор $\vec{n}$ — внешняя нормаль к грани $\mathcal{F}$.

Выход: Новая грань $\mathcal{F}'$ и внешняя нормаль $\vec{n}'$ к ней.

1. Построим аффинный базис ребра $\mathcal{E}$.

   (a) Возьмём произвольную точку $o \in \mathcal{E}$ — начало аффинного базиса. Положим $\mathcal{B}_{\mathcal{E}} = \varnothing$.

   (b) Пока $|\mathcal{B}_{\mathcal{E}}| < d-2$: $\mathcal{B}_{\mathcal{E}} \leftarrow \mathcal{ON}(s_e - o, \mathcal{B}_{\mathcal{E}})$, где $s_e \in \mathcal{E}$. Если получился нулевой вектор в результате ортонормирования, то его не добавляем в базис ребра и переходим к следующей точке.

2. Вычислим вектор $\vec{v}$, перпендикулярный оси вращения и лежащий в плоскости $\mathcal{F}$.

   (a) Возьмём точку $f \in \mathcal{F} \setminus \mathcal{E}$.

   (b) Тогда искомый вектор равняется $\vec{v} = \mathcal{ON}(f - o, \mathcal{B}_{\mathcal{E}})$.

3. Находим точку $s_* \in \mathcal{S}$ такую, что $s_* \notin \mathcal{F}$, и угол между $\vec{v}$ и $\vec{u}$, наибольший среди всех точек роя (то есть $\arccos(\langle \vec{v}, \vec{u} \rangle / \|\vec{u}\|)$ принимает наибольшее значение), где $\vec{u} = \langle s - o, \vec{v} \rangle \vec{v} + \langle s - o, \vec{n} \rangle \vec{n}$ — вектор, перпендикулярный оси вращения и лежащий в новой плоскости, содержащей точку s_*. Соответствующий данной точке s_* вектор $\vec{u}$ обозначим $\vec{r}$.

4. Тогда аффинный базис новой плоскости получается следующим: $(o, \mathcal{B}_{\mathcal{E}} \cup \{\vec{r}\})$.

5. Вычислим внешнюю нормаль $\vec{n}'$ к новой плоскости $\mathcal{F}'$ (см. 4.1).

6. Выполняем построение новой грани $\mathcal{F}'$ на точках роя, попавших в плоскость $\mathcal{L}$, проходящую через ребро $\mathcal{E}$ и точку s (см. процедуру 2.3).

7. Конец процедуры.

Заметим, что нормаль можно вычислять аналогично 4g, но в данном случае количество её пересчётов может быть очень большим, и ошибка накапливающаяся в процессе вычисления уже будет заметно влиять на результат. Поэтому разумно каждый раз нормаль строить точно.

4 Вспомогательные процедуры

4.1 Ориентирование нормали

Вход: $\mathcal{S} \subset \mathbb{R}^d$, вектор нормали $\vec{n}$ и точка p от которой этот вектор отложен.

Выход: Нормаль $\vec{n}'$ такая, что весь рой лежит в неотрицательном полупространстве относительно плоскости с нормалью $\vec{n}'$ и проходящей через точку p.

1. Возьмём очередную точку $s \in \mathcal{S}$. Вычислим скалярное произведение $\langle \vec{n}, s - p \rangle$.

2. Если оно равно нулю, то переходим на 4.1.

3. Если оно положительно, то меняем знак у нормали: $\vec{n}' \leftarrow -\vec{n}$. Иначе нормаль ориентирована верно и $\vec{n}' \leftarrow \vec{n}$.

4. Конец процедуры.

4.2 Вычисление внешней нормали к плоскости

Вход: $\mathcal{S} \subset \mathbb{R}^d$, плоскость $\mathcal{L}$ заданная аффинным базисом $(o, \mathcal{B}_{\mathcal{L}})$.

Выход: Нормаль $\vec{n}$ такая, что весь рой лежит в неотрицательном полупространстве плоскости $\mathcal{L}$.

1. Перебираем орты e_i $(i = 1..d)$, до тех пор пока $\vec{n}_i = \mathcal{ON}(\vec{e}_i, \mathcal{B}_{\mathcal{L}})$ не станет отличным от нуля.

2. Пусть при $i = k$, $\vec{n}_k \neq \vec{0}$. Это искомая нормаль с точностью до ориентации. Выполним переориентирование (см. 4.1). Полученная нормаль будет искомой.

3. Конец процедуры.

4.3 ??? Нужна ли она ??? Процедура получения базиса плоскости, содержащего базис ребра

Вход: базис $\mathcal{B}_{\mathcal{F}}$ $(d-1)$-грани $\mathcal{F}$, $(d-2)$-ребро $\mathcal{E}$ этой грани (важен набор E точек, лежащих в этом ребре).

Выход: базис $\mathcal{B}_{\mathcal{F}}$, содержащий базис ребра $\mathcal{E}$.

1. Выбираем две точки p, p' из E, $p \neq p'$. Точку p полагаем началом аффинного базиса, нормированный вектор $p' - p$ полагаем первым вектором $\vec{b}_1$ конструируемого набора $\mathcal{B}'_{\mathcal{F}}$.

2. Берём точку $p'' \in \mathcal{F}$, $p'' \notin E$. Вектор $p'' - p$, нормированный на фоне $\vec{b}_1$, полагаем $(d-1)$-м вектором $\vec{b}_{d-1}$ конструируемого набора $\mathcal{B}'_{\mathcal{F}}$.

3. Для всех векторов $b \in \mathcal{B}_{\mathcal{F}}$ проверяем, является ли b линейно-независимым на фоне уже накопленного набора $\mathcal{B}'_{\mathcal{F}}$, и, если является, добавляем в $\mathcal{B}'_{\mathcal{F}}$ результат ортонормирования b на фоне текущего набора $\mathcal{B}'_{\mathcal{F}}$.

5 TODO