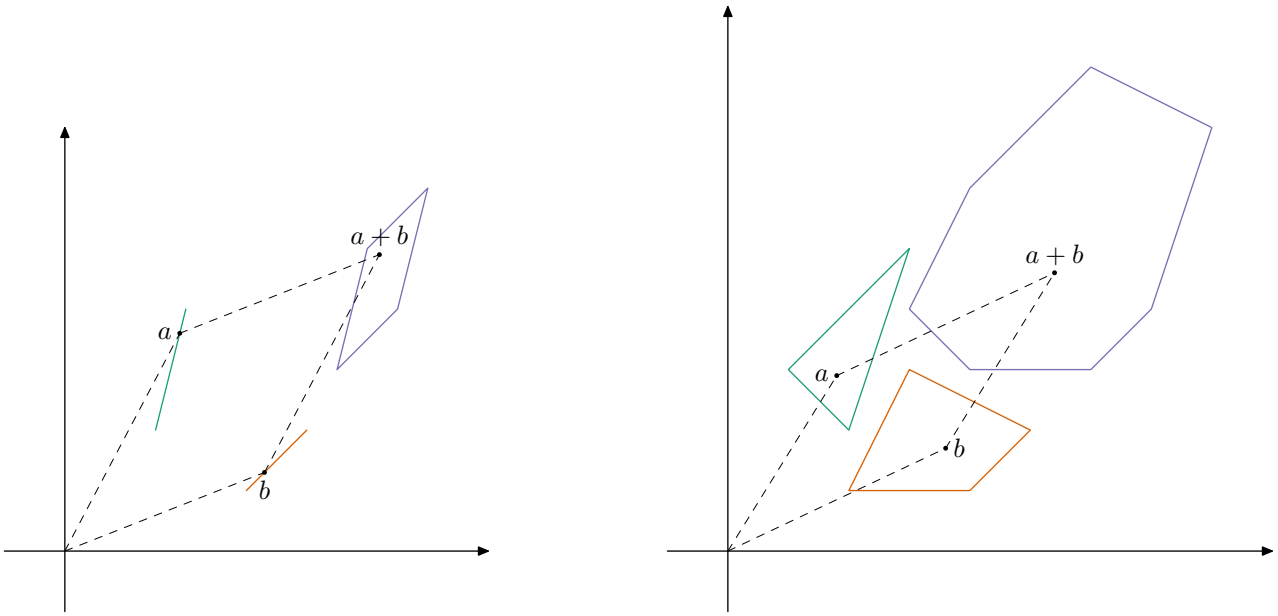
1 Сумма Минковского

1.1 Определение

Суммой Минковского двух множеств A и B называется множество $C = \{a + b : a \in A, b \in B\}$.

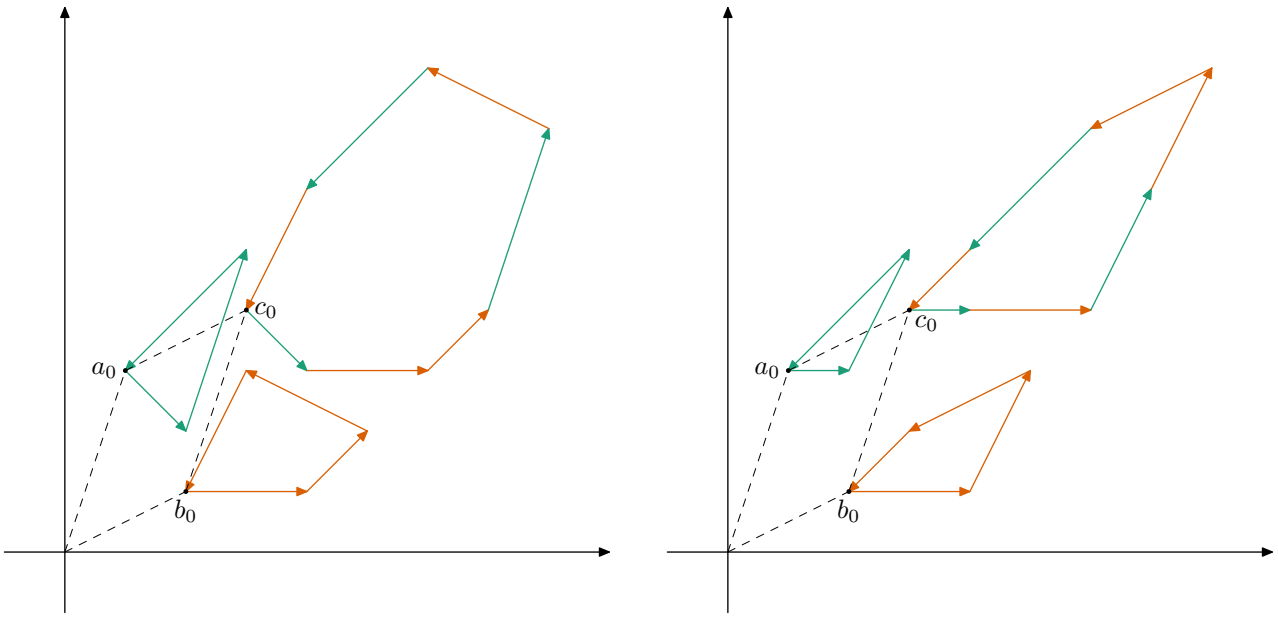


(a) Сумма Минковского двух отрезков (b) Сумма двух двух выпуклых многоугольников

Рис. 1: Примеры суммы Минковского

1.2 Нахождение

Можно доказать, что суммой Минковского двух выпуклых многоугольников является выпуклый многоугольник. Причем, количество сторон в сумме равно сумме количеств сторон у исходных многоугольников. И если ориентировать стороны всех многоугольник в направлении обхода против (или по) часовой стрелке, то множество векторов, получившихся из сторон C, совпадет с объединением множеств векторов, получившихся из сторон A и B. На самом деле, вся предыдущие два предложение не совсем верны. Исключением является случай, когда в множестве векторов сторон A и B есть колинеарные вектора. В таком случае, в множестве сторон C будет находиться сумма этих двух колинеарных векторов. Можно не рассматривать этот случай, как исключение. В таком случае, C может получиться не строго выпуклым, т.е. иметь несколько подрядидущих вершин на одной прямой.



(a) Сумма двух выпуклых многоугольников (b) Пример с колинеарными векторами

Рис. 2: Нахождение суммы Минковского

Отсюда следует алгоритм нахождения суммы двух выпуклых многоугольников. Найдем в каждом многоугольнике вершину, крайнюю в выбраном направлении v. Направление v нужно выбрать таким, чтобы в каждом многоугольнике крайней оказалась одна точка, а не сторона. Например можно выбрать в многоугольниках, минимальные в лексикографическом порядке точки, тогда $v = (-\infty, -1)$. Пусть, в A это будет вершина a_0, а в $B - b_0$. Тогда $c_0 = a_0 + b_0$ является крайней вершиной в C в направлении v. Теперь положим вектора сторон многоугольников A и B в массив, и отсортируем по углу (конечно же, если точки имеют целые координаты, компаратор для сортировки тоже нужно писать в целых числах). И будем откладывать вектора в порядке сортировки, начиная от точки c_0. Причем, если мы рассматриваем обход многоугольников против часовой стрелки, массив отсортированных векторов нужно разрезать по направлению, соответствующему v, повернутому на $90°$ против часовой стрелки.

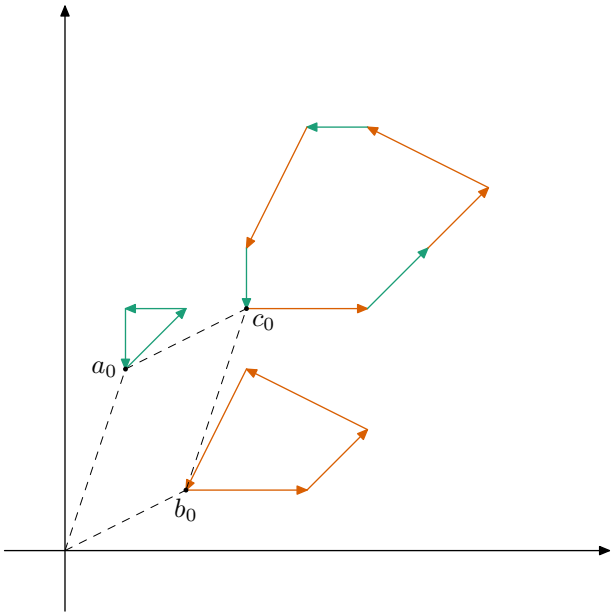


Рис. 3: Если в качестве начальных выбирать точки минимальные лексикографически, то вертикальный вниз вектор должен идти последним в порядке сортировки

1.3 Применение

Сумму Минковского можно применять для проверки пересечения двух выпуклых многоугольников, а также для нахождения расстояния между двумя выпуклыми многоугольниками.

1.3.1 Пересечение

Пусть даны два выпуклых многоугольника A и B, требуется проверить, пересекаются ли они.

$$A \cap B \neq \varnothing \iff \exists p, p \in A, p \in B$$

Обозначим за B' многоугольник B, отраженный относительно начала координат, т.е. каждую координату каждой точки домножили на -1.

$$\exists p, p \in A, p \in B \iff \exists p, p \in A, -p \in B' \iff 0 \in A + B'$$

Таким образом, для проверки пересечения A и B, нужно проверить, лежит ли начало координат внутри суммы Минковского A и B', где $B' - B$, отраженный относительно начала координат.

1.3.2 Расстояние

Обозначим $dist(A, B) = \min\limits_{a \in A, b \in B} |a - b|$. Даны два выпуклых многоугольника A и B, требуется найти $dist(A, B)$.

Как и в пересечении, введем B'.

$$dist(A, B) = \min_{a \in A, b \in B} |a - b| = \min_{a \in A, b \in B'} |a + b| = \min_{c \in A + B'} |c|$$

Значит, чтобы найти расстояние между двумя многоугольниками, нужно найти расстояние от начала координат до суммы Минковского A и B', где $B' - B$, отраженный относительно начала координат. Расстояние от начала координат до выпуклого многоугольника равно минимуму из расстояний от начала координат до его сторон, либо 0, если начало координат лежит внутри многоугольника.

2 Выделение граней планарного графа

Будем использовать структуру под названием DCEL (doubly connected edge list). Вместо каждого ребра, сделаем два полу-ребра, ориентированных в разных направлениях. Два полу-ребра, соответствующие одному и тому же исходному ребру, являются twin-ами друг друга. Так же, для каждой вершины отсортируем полу-ребра, исходящие из нее, по углу. Для полу-ребра e, которое выходит из вершины v, обозначим за $next[e]$ индекс следующего полу-ребра в порядке сортировки вокруг вершины v. Аналогично, за $prev[e]$ обозначим предыдущее полу-ребро в порядке сортировки.

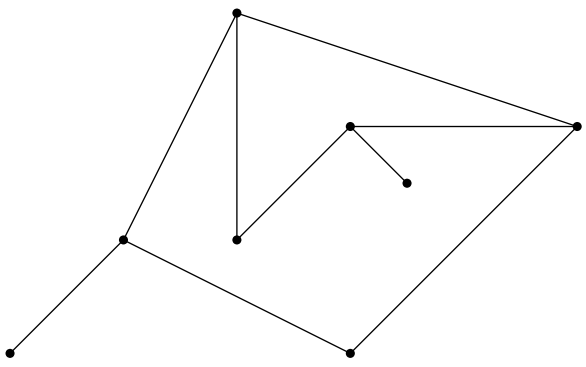


Рис. 4: Исходный планарный граф

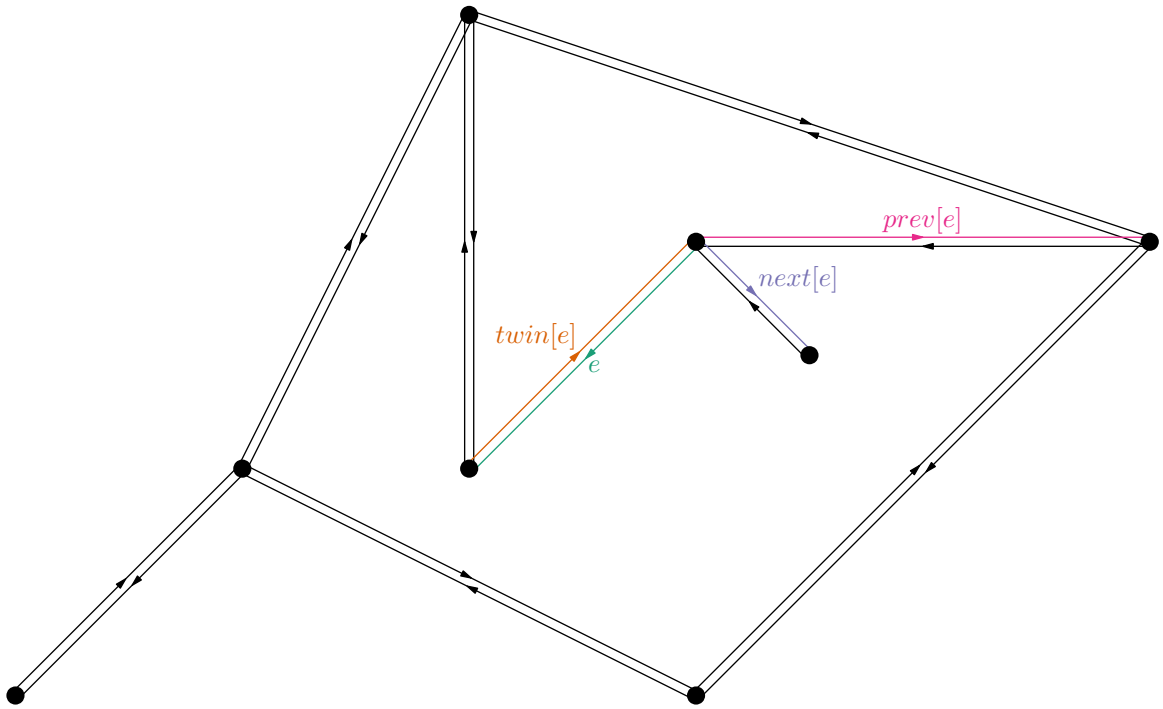


Рис. 5: Построены полу-ребра, определены $twin$, $prev$ и $next$

Возьмем полу-ребро e. Выделим грань, содержащую e. Чтобы перейти к следующему полу-ребру грани, нужно взять $prev[twin[e]]$. Переходим в следующее полуребро, пока не вернемся в исходное.

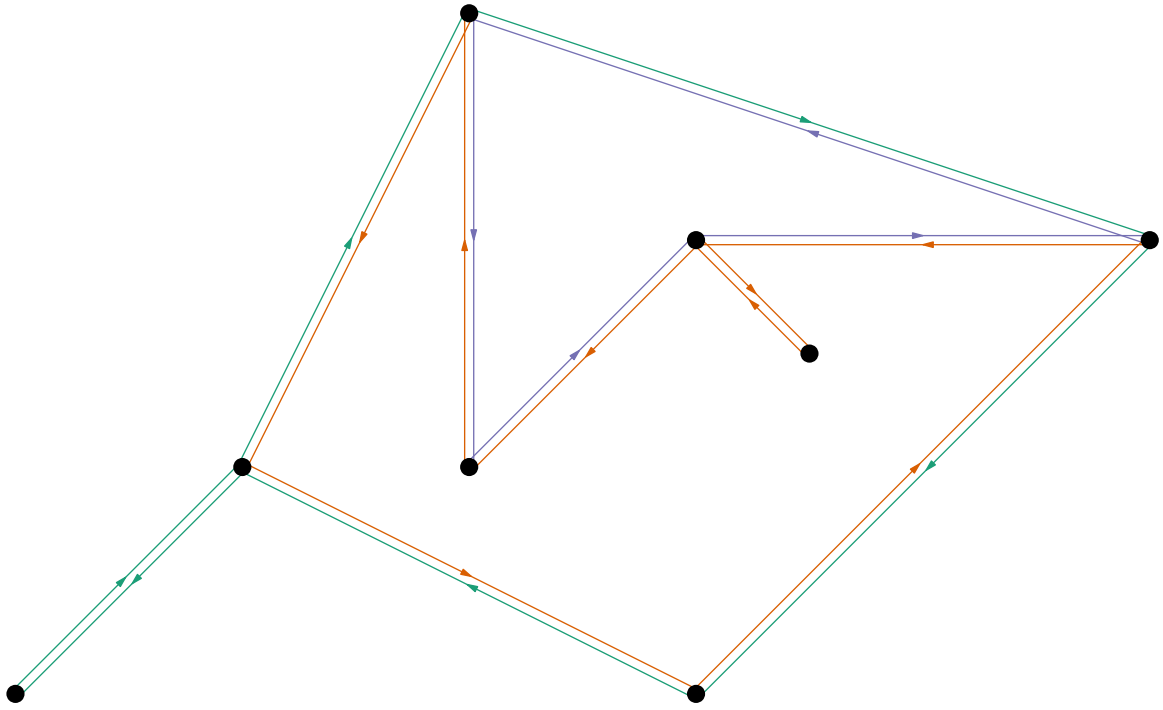


Рис. 6: Выделенные грани