Construction of planar triangulations with minimum degree 5 : Computer Program Part II Version 2.0.1

Rolland Balzon Philippe
Computer Department
Sepro Robotique
85000 La Roche / Yon
France
prolland@sepro-robotique.com

6th June 2002

Abstract

In this article, we describe a computer program based on basic operation to construct a planar triangulations with minimum degree five, denoted $MPG5$. Actually current version doesn't exhaustive, in some sense we know that there exists $MPG5$ such that we doesn't construct. In return at the beginning and during all step we will assume a plan orientation.

Key-Words: Graph Theory, Planar, Triangulation, Maximal, Construction, minimum degree five, computer program

1 Approach

In order to construct an $MPG5$ with $n > 14$ vertices we will use the following sketch:

- Starting by the only graph in $MPG5_{14}$.
- Apply an operation called T which increment n by adding one vertex. This operation conserving all properties : planarity, triangulation, degree minimal, plan orientation and increment number of vertices by adding one vertex.
- Apply d time a flip, diagonal operations (between one and a fixed variable max_D). And this for each order between 14 and n. Again we have assuming following conservations : planarity, triangulation, degree minimal, plan orientation.

2 Definitions

Definition 1 *Let $G = (X, E)$ be an $MPG5_{n>14}$ and x with $dg(x) > 5$. We describe $N(x)$ in clockwise order by $\{x_1, \ldots, x_k, \ldots, x_q\}$ where $k \in [4, q-2]$ and $q > 5$. A such path is denoted by $[x; x_1, x_k]$. See figure 2.*

Definition 2 *$T[x; x_1, x_k]$ is the graph G after the explosion of x in two new adjacent vertices x', x'' such that in clockwise order $N(x') = \{x_1, x_2, \ldots, x_{k-1}, x_k, x''\}$ and $N(x'') = \{x_1, x', x_k, x_{k+1}, \ldots, x_q\}$. See figure 2.*

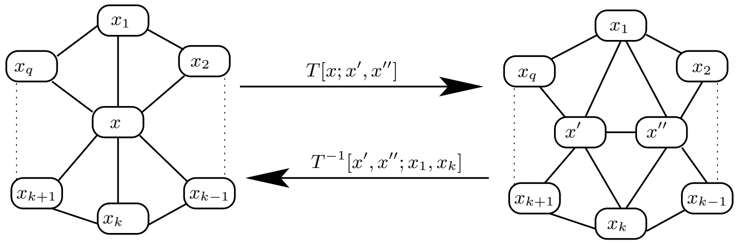

Figure 2. T and T^{-1} transformations.

Definition 3 *Let $G = (X, E)$ be an $MPG5$. We denote*

$$X_{sup6} = \{x \in X / dg(x) \geq 6\},$$
$$X_{inf6} = \{x \in X / dg(x) \leq 5\}.$$

3 About explosion

Program is defined by a main loop between 14 and n (ie number of vertices), each step add a vertex by T. Inside the main loop, first select a vertex x with at least six neighbors. Secondly, we have to select two neighbors of x called x_1 and x_k such that $k \in [4, q-2]$. Third, dispatch and update neighborhood.

Algorithm 4

TITLE: $MPG5_n$ `construction programming`
INPUT: n `number of vertices`
OUTPUT: `At random` $MPG5$ `with` n `vertices`
$n_i = 14$; $G =$ `Load` $(MPG5_{14})$;
While $n_i \leq n$ **do**
  $x =$ `random`$(X_{sup6}, 1, |X_{sup6}|)$;
  $x_1 =$ `random`$(N(x), 1, dg(x) - 3)$;
  $x_k =$`random`$(N(x),$`position`$(N(x), x_1, 3)$, `position`$(N(x), x_1, -3)$;
  $N(x) = \{x_1, \ldots, x_k, n_i + 1\}$ *in clockwise order*
  $N(n_i + 1) = \{x_k, x_{k+1}, \ldots, x_1, x\}$ *in clockwise order*
  ForAll *vertices y in* $\{N(n_i + 1) - x - x_1 - x_k\}$ ***Do***
    In $N(y)$ Replace x by $n_i + 1$
  End ForAll
  In $N(x_1)$ add $n_i + 1$ after x
  In $N(x_k)$ add $n_i + 1$ before x
  $n_i = n_i + 1$
End While

3.1 x_1, x_k choices

We looking a couple such that always we have `position`$(x,x_1) <$ `position` (x,x_k), where `position`(v,i) function are giving index position of i in $G[v]$ array. In this way without restriction, we will simplify computer program.

3.2 Details

`position`(N,v,α) This function give position of a vertex v in a neighborhood N with an offset α modulo $|N|$.

Example: In algorithm 4, Let $x = 1$ and $N(x) = \{2, 3, 4, 5, 6, 7, 8, 9\}$. Possibilities for x_1 are $\{2, 3, \ldots, 6\}$. We are considering different choices :

- Let $x_1 = 6$. Possibilities for x_k are only 9.
- Let $x_1 = 2$. Possibilities for x_k are $\{5, 6, 7\}$.

- Let $x_1 = 3$. Possibilities for x_k are $\{4, 5, 6, 7, 8\}$.

4 About flip

We looking for a vertex a in X_{sup6} only for graph with an order at least 16. Around the first neighborhood of a we are looking for the list of all vertices in X_{sup6}. We will describe these cases:

- All neighborhood of a are in X_{inf6}, so we looking for another vertex a.
- Let $k = |N(a) \cap X_{sup6}| > 1$. We have to selecting at random a neighbor b between these k possibilities.

Let a the first selection in X_{sup6} and b in $N(a) \cap X_{sup6}$. We are denoting c and d the two common neighbors of a, b.

We have to replace the following neighborhood :

- Around a.
  - Before $D(a, b)$: $\ldots, c, b, d, \ldots$
  - After $D(a, b)$: $\ldots, c, d, \ldots$
- Around b.
  - Before $D(a, b)$: $\ldots, d, a, c, \ldots$
  - After $D(a, b)$: $\ldots, d, c, \ldots$
- Around c.
  - Before $D(a, b)$: $\ldots, b, a \ldots$
  - After $D(a, b)$: $\ldots, b, d, a \ldots$
- Around d.
  - Before $D(a, b)$: $\ldots, a, b \ldots$
  - After $D(a, b)$: $\ldots, a, c, b \ldots$

5 About X_{sup6}

We have initializing X_{sup6} with $MPG5_{14}$ i.e. the starting graph. After each operation flip or explosion we have updated X_{sup6} by using two functions.

- Firstly one called `Xsup6 add vertex` which add a vertex x if we have the following condition: ((`new vertex`(x) AND (`degree`$_{before}$(x)$>$5)) OR (`degree`$_{before}$(x)$==$5) AND (`degree`$_{after}$(x)$>$5))
- Secondly one called `Xsup6 del vertex` which remove a vertex x if $dg(x) = 5$.

We'll detail these using functions in explosion and flip operations.

- Explosion :
  - `Xsup6 add vertex`($v1$,NOT `new vertex`)
  - `Xsup6 add vertex`($v2$,NOT `new vertex`)
  - `Xsup6 add vertex`(N,`new vertex`)
  - `Xsup6 del vertex`(v)

- Flip:
  - `Xsup6 add vertex`(c,NOT `new vertex`)
  - `Xsup6 add vertex`(d,NOT `new vertex`)
  - `Xsup6 del vertex`(a)
  - `Xsup6 del vertex`(b)

6 Data structure

Actually, we have using a simple matrix G such that for all vertex x we have defining neighborhood in clockwise order see following algorithm.

Algorithm 5

TITLE: `Load initial graph in Data structure`
INPUT: $MPG5_{14}$
OUTPUT: G `Data structure`
For *($i = 1$; $i \leq 14$; $i++$)* **do**
    For *($j = 1$; $i \leq dg(i)$; $j++$)* **do**
        G[i][j]= j th neighbor of vertex i in clockwise order
    End For*;*
End For

7 Checking

In order to verifying plan orientation either each step during 14 and n or only on the final stage, we have programmed an orientation checking, here we are detailing this checking.

Let x a vertex, and x_i, x_{i+1} two consecutive neighbors of x in clockwise order. Algorithm 6 will check :

- there exists x_{i+1}, x two consecutive neighbors of x_i in clockwise order.
- there exists x, x_i two consecutive neighbors of x_{i+1} in clockwise order.

This will be check for all vertex x and all two consecutive neighbors of x.

Algorithm 6

TITLE: `Plan orientation checking`
INPUT: $MPG5$ `data structure with` $n > 14$ `vertices`
OUTPUT: `boolean`
For *($x = 1$; $x \leq n$; $x++$)* **do**
    For *($i = 1$; $i < dg(x)$; $i++$)* **do**
        $x_i = G[x][i]$;$x_{i+1} = G[x][i+1]$
        `check`*() = looking for* x_{i+1}, x *around* x_i
        `check`*(*x_{i+1}*) = looking for* x, x_i *around* x_{i+1}
        If *(`check`(*x_i*)==TRUE)* ***AND*** *(`check`(*x_{i+1}*)==TRUE))* ***THEN*** *continue*
            Else *print "Wrong Orientation around (*x,x_i,x_{i+1}*)"; Exit*
    End For*;*
End For