Hang Lung Mathematics Awards
Vol. 5 (2012)

Hang Lung Mathematics Awards 2012

Bronze Award

Complexity Reduction of Graph

Team member: Tsam Kiu Pun
Teacher: Mr. Cyril Lee
School: St. Mary's Canossian College

Hang Lung Mathematics Awards
Vol. 5 (2012), pp. 177–188

COMPLEXITY REDUCTION OF GRAPH

TEAM MEMBER
Tsam Kiu Pun

TEACHER
Mr. Cyril Lee

SCHOOL
St. Mary's Canossian College

Abstract. Many real-world problems can be modeled mathematically as graphs. Some of these graphs are complex because of their large numbers of vertices and edges. To develop applications over any of these graphs, a graph which is less complex but having characteristics similar to the original graph will always be very useful. We propose in this report a new graph reduction method by performing a singular value decomposition on the adjacency matrix of a complex graph. We also propose a notion of loop decomposition which is a generalization of graph triangulation, from which we also derive a measure of graph complexity.

1. Introduction

Complex networks such as the neural network in a human brain and the road distribution in a country can be modeled mathematically as graphs with large number of vertices and edges. Another well-known example is the Internet which connects a huge number of websites. In this latter case we have the so-called "Webgraph". This kind of complex graphs have been studied from many different angles depending on applications ([7], [8]), one of them is to find another graph which can retain the basic properties of the original graph but is less complex in terms of the number of edges and/or vertices. This is commonly called "graph reduction" or "complexity reduction of graphs". There are several existing techniques for tackling this problem, for example, clustering and segmentation [6], ideas commonly used in computer science. In this report we propose yet another approach to graph reduction which we believe to be original. The method is based on two concepts one is related to the topology of a graph and the other the approximation of a matrix. In particular we will introduce a notion of "loop decomposition" with graphs of which triangulation can be considered as a special case. Then we propose the use of singular value decomposition to find an adjacency matrix which approximates

that of the original one. This new adjacency matrix, having fewer "lower-order loops", corresponds to a graph having lower complexity as calculated by a formula defined along with loop decomposition.

The remaining of our report is divided into five sections. In Section 2 we give the background knowledge of graph theory used in this study. We then introduce our generalization of triangulation in Section 3. The idea is further expanded in Section 4 where a measure of graph complexity is defined. We then introduce singular value decomposition in Section 5 together with a graph reduction algorithm. An example is used to illustrate the method. In Section 6 we give a brief conclusion of our work and suggest some future directions.

***Acknowledgement*:** The author of this report would like to express her sincere gratitude to Mr. Cyril Lee for his support and to Professor C.P. Kwong for his assistance. Professor Kwong not only suggested to the author this interesting topic, but he was also lavish in rendering his valuable advice when the author was obstructed by problems on the course of her research.

2. Graph Theory Background

Though the following background knowledge of graph theory are really elementary, the reasons of inclusion herewith are their relevance to our present study and for ease of reference.

Let $G = (V, E)$ be an ***undirented graph*** where V is a finite nonempty set of ***vertices*** and E is a set of unordered pairs of distinct vertices $p, q \in V$, called the ***edges*** of G. The graphs we are going to study are all undirected graphs. For simplicity we just call them graphs in the sequel unless specified.

A graph can be depicted as a diagram with points as vertices and lines joining the vertices as edges. On the other hand, we can also represent the graph by an $n \times n$ ***adjacency matrix*** A where n is the number or vertices. Every element of A assumes the value of either 0 or 1. If A_{ij}, the element at the i-th row and j-th column of A, is 1, then there is an edge connecting the i-th vertex and the j-th vertex. Otherwise the value of this matrix element is 0, indicating no connection between these two vertices. We assume that there is no self-connecting vertex

or multiple edges between two vertices. Clearly A is symmetric. Here is an example:

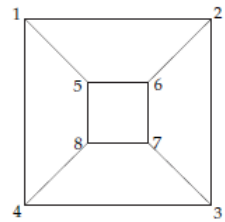

FIGURE 1. A graph and its adjacency matrix.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Definition 1. *Let $G = (V, E)$ be graph. A **walk** is a sequence of vertices and edges of G of the form*

$$\{v_1, \{v_1.v_2\}, v_2, \{v_2, v_3\}, v_3, \ldots, v_{n-1}, \{v_{n-1}, v_n\}, v_n\},$$

or simply

$$\{v_1, v_2, v_3, \ldots, v_{n-1}, v_n\}.$$

*A walk is **closed** if $v_1 = v_n$. A closed walk with at least three distinct vertices (except for the first and the last) is called a **cycle**. A cycle with n edges is termed an n-**cycle**. It is a **triangle** when $n = 3$.*

To assist our following discussion we shall use

$$G \to A(G)$$

to denote the process of constructing the adjacency matrix $A(G)$ associated with a graph G. Similarly, when we write

$$A \to G(A),$$

we are referring to the process of drawing the graph $G(A)$ corresponding to an adjacency matrix A.

3. Triangulation with Graphs and Its Generalization

We may define complexity of a graph G based on the number of triangles it possesses. This consideration leads naturally to the concept of triangulation in graph theory. The following notions are well known [4].

Definition 2. *A graph is said to be* ***embedded*** *in a surface S if, when it is drawn on S, no two edges of G intersect. In case the surface is the plane, such a graph is called* ***planar****. Furthermore, a planar graph G is named* ***maximally planar*** *if, for every pair of nonadjacent, distinct vertices u and v, the graph $G + \{u, v\}$ is nonplanar.*

A graph embedded in a plane divides the plane into areas called ***faces***. If the graph is maximally planar, then it is clear that every face is bounded by a 3-cycle, i.e., a triangle. We say the plane embedding a ***triangulation***, The following is an example.

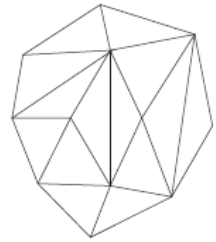

FIGURE 2. Triangulation.

Consider removing some inner edges (dashed lines in Fig. 3) of the graph in Fig. 2, we obtain 4-cycles as shown in Fig. 3. Note that in these 4-cycles there is no edge connecting two nonadjacent vertices.

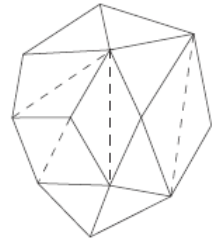

FIGURE 3. 4-cycles obtained from removing edges.

We give this kind of 4-cycles a special name of 4-***loops*** to distinguish them from 4-cycles containing triangles. In the following example, $\{\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4, \hat{v}_1\}$ is a 4-cycle whereas $\{v_1, v_2, v_3, v_4, v_1\}$ is a loop.

We may consider a 4-loop as a generalization of a triangle for embedding a graph in a surface. This generalization can be further extended to higher-order cycles as follow.

Definition 3. *Let $G = (V, E)$ be a graph. An n-****loop*** *is an n-cycle $\{v_1, v_2, v_3, \ldots, v_{n-1}, v_n, v_1\}$, $n \geq 3$, in which the number of edges connecting v_i and v_j, $j \neq i + 1$*

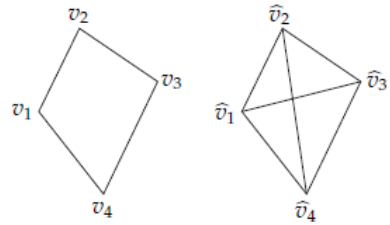

FIGURE 4. 4-loops are different from 4-cycles.

for every $i = 1, 2, \ldots, n-1$, *is zero. An exception is the edge* $\{v_1, v_n\}$. *We denote an n-loop by* L^n.

Hence a triangle is a 3-loop in our new definition. We show in the following diagram loops from L^3 to L^7. Note in particular that **the face bounded by any** $(n > 3)$**-loop contains no triangle.**

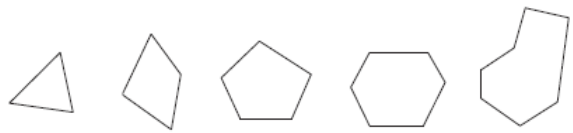

FIGURE 5. From L^3 to L^7

Suppose we may remove edges from the triangulation in Fig. 2. Then we obtain loops of higher orders $(n > 3)$ as shown in Fig. 6

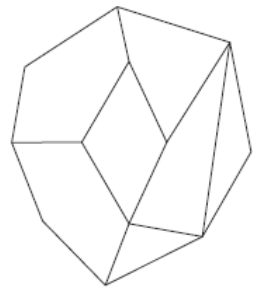

FIGURE 6. Triangulation becomes loop decomposition.

4. Loop Decomposition and Graph Complexity

Fig. 6 is an example we would like to give a name of "loop decomposition" for plane embedding. This is in contrast with the usual triangulation. Furthermore,

the graph in Fig. 6 is deemed to be simpler than the graph in Fig. 2 if complexity is measured as some monotonically increasing function of the number of loops, with high-order loops assigned lower complexity. We will formally formulate this model of complexity shortly, which serves to measure the effectiveness of the graph reduction method we will propose subsequently. In any case we may gain useful insights through the following observation. Since a 3-loop (a triangle) is the most basic component in enclosing a face, and a 4-loop admits decomposition into two 3-loops and a 5-loop admits decomposition into three 3-loops, we see a higher degree of reduction in complexity when extracting a higher-order loop than getting a lower-order loop through removal of edges. For instance, consider the reduction of the graph in Fig. 2 or Fig. 6 to a single 9-loop comprising just the nine outer vertices.

We may use the following two different methods to define a measure of the contribution of an n-loop to the overall complexity of a given graph.

- **Method A:** For every n-loop L^n we add edges to its vertices to turn L^n into a complete graph K^n. Since a triangle can be singled out by choosing three vertices from the total n vertices, the number of triangles so obtained is $\Delta(K^n) = \binom{n}{3}$. The inverse of this number, labeled $\|L^n\|_A$:
$$\|L^n\|_A = \frac{1}{\Delta(K^n)} = \frac{1}{\binom{n}{3}}, \tag{1}$$
is called the ***complexity of the n-loop*** (under **Method A**).
- **Method B:** For every n-loop L^n we choose an arbitray vertex v_i and connect v_i to every other vertices. Denote the resulting graph as H^n. We see that $\Delta(H^n) = n - 2$. The inverse of this number, labeled $\|L^n\|_B$;
$$\|L^n\|_B = \frac{1}{\Delta(H^n)} = \frac{1}{n-2}, \tag{2}$$
is called the ***complexity of the n-loop*** (under **Method B**).

Sometimes we simply write $\|L^n\|$ under a context where the choice of method is irrelevant. For example, for either **Method A** or **B**, we have
$$\|L^3\| = 1, \text{ and } \|L^3\| > \|L^4\| > \|L^5\| \cdots > \|L^n\|. \tag{3}$$

Remark 4. *The graph K^n in **Method A** is nonplanar. That means the notion of loop decomposition may be generalized to the study of nonplanar graphs.*

[See reviewer's comment (2)]
It is interesting to plot $\|L^n\|_A$ and $\|L^n\|_B$ for various values of n, as follow.

Definition 5. *Let $G = (V, E)$ be a graph with n vertices. Then the **complexity of G**, denoted $\|G\|$, is given by*
$$\|G\| = |L^3|\|L^3\| + |L^4|\|L^4\| + \cdots + \|L^n|\|L^n\|, \tag{4}$$

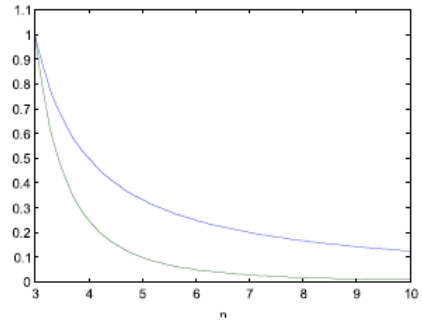

FIGURE 7. $\|L^n\|_A$ (lower curve) and $\|L^n\|_B$ (upper curve).

where $|L^i|$ and $\|L^i\|$ is, respectively, the number of i-loops in G and the complexity of the i-loop, $i = 3, 4, \ldots, n$. Note that L^i may have value zero for some i.

5. Singular Value Decomposition and Complexity Reduction of Graphs

Singular value decomposition (SVD) is an important result in applied linear algebra as it provides a theoretical basis for matrix approximation and the computation of generalized inverse ([1],[9]). Our proposed method also makes use of the fundamental properties of SVD, stated in the two theorems below, for effective graph reduction.

Theorem 6. *Let A be an $m \times n$ real matrix. Then there exists a factorization*

$$A = U\Sigma V^T \tag{5}$$

where U is an $m \times m$ real orthonormal matrix, V^T is the transpose of an $n \times n$ real orthonormal matrix V, and Σ is an $m \times n$ diagonal matrix with nonnegative real elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$: [See reviewer's comment (3)]

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & \vdots & \\ & \sigma_2 & & & \vdots & \\ & & \ddots & & \vdots & 0 \\ & & & \sigma_r & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & 0 & & \vdots & 0 \end{bmatrix}, \tag{6}$$

where r is the rank of A

Given a graph G, we can construct its adjacency matrix $A(G)$. As stated in Section 2, we write $G \to A(G)$ to represent this process. Now we can find the SVD of $A(G)$ and obtain the set of singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Suppose we set the

last k singular values $\sigma_{r-k+1}, \sigma_{r-k+2}, \cdots, \sigma_r$ to zero to obtain a new singular value matrix $\tilde{\Sigma}$. Then replace Σ in (5) by $\tilde{\Sigma}$, we obtain a rank-$(n-k)$ matrix $\widetilde{A(G)}$ which is an approximation of $A(G)$. The error of approximation is given by the following theorem [1].

Theorem 7. *Let A be an $m \times n$ real matrix and $U\Sigma V^T$ is its singular value decomposition. Let $\tilde{\Sigma}$ be the matrix obtained by setting $\sigma_{r-k+1}, \sigma_{r-k+2}, \cdots, \sigma_r$ in Σ to zero. Then the matrix $\tilde{A} = U\tilde{\Sigma}V^T$ is the best rank-$(n-k)$ approximation of A in the sense that the Frobenius norm of the error $A - \tilde{A}$:*

$$\|A - \tilde{A}\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} - \tilde{a}_{ij})^2 \right)^{\frac{1}{2}}, \tag{7}$$

is minimized.

Since $\widetilde{A(G)}$ is in general not in the form of an adjacency matrix though $\widetilde{A(G)}$ is "closed" to $A(G)$, we cannot draw a graph directly from this matrix. In any case, we can approximate $\widetilde{A(G)}$ further to obtain a matrix $\widehat{A(G)}$ which is an adjacency matrix. The approximation is based also on Frobenius norm minimization. The following steps are precise descriptions of the approximation algorithm.

- **Step 1.** Set the diagonal of $\widetilde{A(G)}$ to zero. Also set to zero the elements of $\widetilde{A(G)}$ corresponding to null edges in the original $A(G)$.
- **Step 2.** Compute $p_{ij} = \widetilde{A(G)}_{ij}^2 + \widetilde{A(G)}_{ji}^2$ for all i, j. If p_{ij} is smaller than a threshold T determined by

$$T = \tau[max(p_{ij}) - min(p_{ij})] + min(p_{ij}) \quad \text{for all } i, j, \tag{8}$$

  set $\widetilde{A(G)}_{ij} = \widetilde{A(G)}_{ji} = 0$. In this formula for calculating the threshold T, $0 < \tau \leq 1$ is an error control parameter.
- **Step 3.** Set all the remaining elements of $\widetilde{A(G)}$, except its diagonal, to one. The resulting matrix, labeled $\widehat{A(G)}$, is an adjacency matrix.

We denote the graph corresponding to $\widehat{A(G)}$ as $\hat{G}$, which is the desired reduced graph of the original graph G. Our graph reduction algorithm can be summarized as a sequence of transformations:

$$G \to A(G) \to \widetilde{A(G)} \to \widehat{A(G)} \to \hat{G}, \tag{9}$$

which is best illustrated by an example.

Example 8. *Fig. 8 shows a graph G with eight vertices and ten 3-loops.*

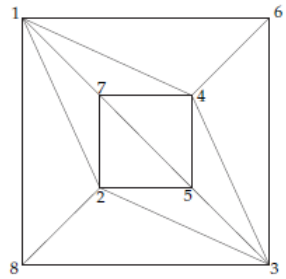

FIGURE 8. Graph G with ten 3-loops.

The adjacency matrix $A(G)$ associated with G is

$$A(G) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After performing the SVD of $A(G)$ we obtain the set of singular values as $\{\sigma_1 = 4.3680, \sigma_2 = 2.3865, \sigma_3 = 1.8588, \sigma_4 = 1.6180, \sigma_5 = 1.0000, \sigma_6 = 1.0000, \sigma_7 = 0.8773, \sigma_8 = 0.6180\}$.
Discarding the last two smallest singular values $\sigma_y = 0.8773$ and $\sigma_8 = 0.6180$, we obtain the rank-6 approximation $\widetilde{A(G)}$ of $A(G)$ as

$$\widetilde{A(G)} = \begin{bmatrix} -0.2726 & 1.0182 & 0.1746 & 1.0182 & 0.2406 & 0.9091 & 0.9642 & 0.9091 \\ 1.0182 & -0.0068 & 1.0182 & -0.0068 & 0.9619 & 0.0338 & 0.9619 & 1.0338 \\ 0.1746 & 1.0182 & -0.2726 & 1.0182 & 0.9642 & 0.9091 & 0.2406 & 0.9091 \\ 1.0182 & -0.0068 & 1.0182 & -0.0068 & 0.9619 & 1.0338 & 0.9619 & 0.0338 \\ 0.2406 & 0.9619 & 0.9642 & 0.9619 & -0.2996 & 0.1901 & 0.8712 & 0.1901 \\ 0.9091 & 0.0338 & 0.9091 & 1.0338 & 0.1901 & -0.1687 & 0.1901 & -0.1687 \\ 0.9642 & 0.9619 & 0.2406 & 0.9619 & 0.8712 & 0.1901 & -0.2996 & 0.1901 \\ 0.9091 & 1.0338 & 0.9091 & 0.0338 & 0.1901 & -0.1687 & 0.1901 & -0.1687 \end{bmatrix}$$

Then we perform the preceding 3-step algorithm to transform $\widetilde{A(G)}$ into $\widehat{A(G)}$. In Step 2 of this algorithm we get

$$max(p_{ij}) = 2.1375 \quad \text{and} \quad min(p_{ij}) = 1.5181.$$

Setting $\tau = 0.3$ gives

$$\begin{aligned} T &= \tau[max(p_{ij}) - min(p_{ij})] + min(p_{ij}) \\ &= 0.3(2.1375 - 1.5181) + 1.5181 \\ &= 1.7039. \end{aligned}$$

The final Step 3 results in the matrix $\widehat{A(G)}$*:*

$$\widehat{A(G)} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

from which we obtain the reduced graph $\hat{G}$ *as shown in Fig. 9.*

Observe that $\hat{G}$ *has four 3-loops and three 4-loops. By using Definition 5 for graph complexity, we have (using* ***Method A****)*

$$\begin{aligned} \|\hat{G}\| &= |L^3|\|L^3\|_A + |L^4|\|L^4\|_A \\ &= 4(1) + 3(0.25) \\ &= 4.75 \ . \end{aligned}$$

This is compared with the complexity of the original graph G*:*

$$\begin{aligned} \|G\| &= |L^3|\|L^3\|_A \\ &= 10(1) \\ &= 10 \ . \end{aligned}$$

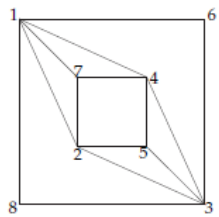

FIGURE 9. Reduced graph $\hat{G}$ with four 3-loops and three 4-loops.

In computing this example we use the free software FreeMat which is a reduced version of MATLAB.

6. Conclusion and Future Work

We have documented in this report our research on the problem of reducing the complexity of a graph. Two key ideas have been involved in this study, namely the loop decomposition of graphs, which is topological in nature, and the low-rank approximation of the adjacency matrix via singular value decomposition, which

is analysis inclined. Though we have only discussed graphs without weightings attached to edges, our method should be applicable also to weighted graphs. Another possible direction of future research is to study whether we can apply singular value decomposition to the "Laplacian" of a graph–another matrix representation of graphs ([2],[3]). In doing so we will be entering the realm of algebraic graph theory [5].

REFERENCES

[1] Ben-Israel A. and Greville N.E., *Generalized Inverses: Theory and applications*, 2nd ed., Springer-Verlag, New York, 2003.
[2] Chung F.R.K., *Spectral Graph Theory*, American Mathematical Society, 1997.
[3] Cvetković D.M., Rowlinson P., and Simić S., *An Introduction to the Theory of Graph Spectra*, Cambridge University Press, 2010.
[4] Foulds L.R., *Graph Theory Applications*, Springer-Verlag, New York, 1992.
[5] Godsil C. and Royle G., *Algebraic Graph Theory*, Springer-Verlag, New York, 2001.
[6] Lermé N., *Reducing graphs in graph cut segmentation*, 17 th IEEE International Conference on Image Processing (2010).
[7] Matthias D. (ed.), *Structural Analysis of Complex Networks*, Springer Science & Business Media, New York, 2011.
[8] Reuven C., *Complex Networks: Structure, robustness, and function*, Cambridge University Press, 2010.
[9] Strang G., *Linear Algebra and Its Applications*, 4th ed., Brooks Cole, 2005.

Reviewer's Comments

Grammatical mistakes and typos

1. The reviewer has comments on the wordings, which have been amended in this paper.
2. Remark: The graph K^n in Method A is nonplanar for $n > 4$.
3. $m \times n$ rectangular diagonal matrix

Comments

The paper is about complexity reduction of graphs. The author made use of tools from graph theory and linear algebra to simplify a possibly large and complicated graph. The research problem itself is important and has a lot of applications.

The paper is well-organized. It starts with some motivations of the project, then review of background materials and a proposed algorithm for complexity reduction. An example was given at the end to illustrate the graph reduction method and the computation of complexity. One thing could be added to this example is to show how the complexity depends on the error control parameter τ picked. More examples may also be given.

While the proposed algorithm reduces the complexity of a graph, the author may further investigate whether the algorithm preserves certain important characteristics of the original graph.

Some of the definitions and statements in the paper are not very precise. For example, at the end of section 2, an association between graphs and adjacency matrices was given. Strictly speaking, in order to do that, an ordering on the vertices is needed. Also, the graph in figure 2 is not maximally planar. Is maximally planar a condition needed for being a triangulation?

According to definition 3, the cycles $\{1, 6, 3, 8, 1\}$ and $\{1, 6, 3, 5, 7, 1\}$ in the graph of figure 8 are both loops. However, they do not appear in the computation of the complexity. Similarly, in definition 5, it should be noted that complexity should be defined on embedded graphs. It means that $\|G\|$ depends not only on the abstract data of vertices and edges of G, but also how G is drawn on the plane. The author should be careful about these definitions and statements.