Interactive Vertex Coloring of Polyhedral Graphs

Hidetoshi Nonaka

Graduate School of Information Science and Technology,
Hokkaido University, N14W9, Sapporo 060 0814, Japan
nonaka@main.ist.hokudai.ac.jp

Abstract

The author has developed an interactive learning system of polyhedra, based on graph operations and simulated elasticity potential method, mainly for educational purpose. In this paper, we introduce a learning subsystem of vertex colouring, edge colouring, and face colouring, based on minimum spanning tree and degenerated polyhedron.

Keywords: Vertex Colouring, Polyhedral Graph, Simulated Elasticity, Internalist and Externalist Anticipation, Degenerated Polyhedron.

1 Introduction

Since 2004, we have developed an interactive learning system of polyhedra, mainly for educational purpose [1-5]. The modelling and forming polyhedra is based on graph operations and simulated elasticity potential method. The interaction is promoted by internalist anticipation and externalist anticipation [6]. By using this system, the user or the learner can make and handle various polyhedra, including Platonic solids, Archimedean solids [3], Kepler-Poinsot solids [2], fullerenes molecular structures, and geodesic dome constructions.

In this paper, we introduce a learning subsystem of vertex colouring, edge colouring, and face colouring, based on minimum spanning tree and degenerated polyhedron. Vertex colouring of polyhedral graph itself is trivial in a mathematical sense, and it is not novel also in a practical sense. However, visibility and interactivity can be helpful for the user to understand intuitively the mathematical structure and the computational scheme, by visualizing the process of the calculation, and by allowing the user to anticipate the result and to contribute the computation.

2 Interactive Modelling System of Polyhedra

In this section, we summarize the system of interactive modelling of polyhedra described in [1-5]. It consists of three subsystems: graph input subsystem, wire-frame subsystem, and polygon subsystem. In these subsystems, various types of internalist anticipation and externalist anticipation are induced. The system has been developed with C++ Builder on MS-Windows XP and MS-Windows 7.

International Journal of Computing Anticipatory Systems, Volume 28, 2014
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-17-2

2.1 Graph Input Subsystem

The first step of the modelling of polyhedron is drawing a polyhedral graph isomorphic to the anticipated polyhedron. Figure 1 shows a screen shot of graph input subsystem, where a graph isomorphic to icosahedron is drawn. In the subsystem, vertex addition, vertex deletion, edge addition, and edge deletion are implemented as fundamental operations. Some additional utilities are also implemented such as grid lines, grid snapping, vertex colouring according to degrees, and so on. The process of graph input and interaction in this subsystem are mainly based on internalist anticipation of the learner [6].

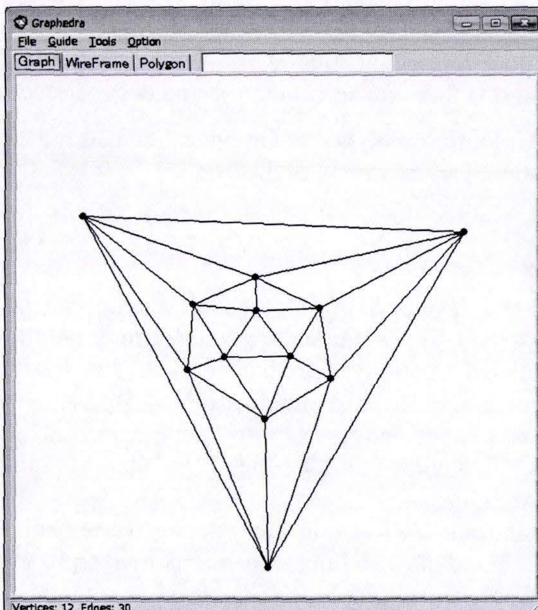

Figure 1: Screen shot of Graph Input Subsystem.

2.2 Wire-Frame Subsystem

After constructing a polyhedral graph the next step is arranging vertices in 3D space with virtual springs and Hooke's law. Figure 2 shows a screen shot of wire-frame subsystem. Wire-frame polyhedron can be formed by controlling the natural length of virtual spring corresponding to binary relations. The diameter of a graph $G = (V, E)$ is defined by (1), where the length of shortest path from u to v is denoted by $length(u, v)$.

$$dmr(G) \triangleq \max_{u,v \in V} \left(length(u,v) \right) \tag{1}$$

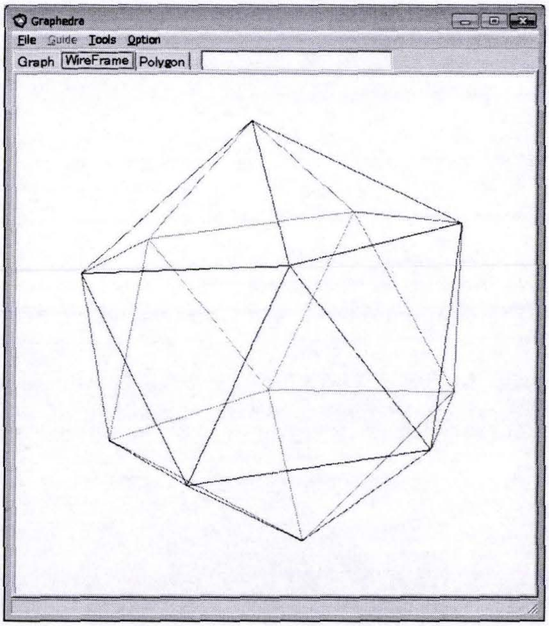

Figure 2: Screen shot of Wire-frame Subsystem.

We define following three types of binary relations between vertices.

$$adjacent(u,v) = \begin{cases} true & (u,v) \in E \\ false & otherwise \end{cases} \tag{2}$$

$$neighbour(u,v) = \begin{cases} true & (u,v) \notin E \wedge \exists w.\left[(u,w) \in E \wedge (v,w) \in E\right] \\ false & otherwise \end{cases} \tag{3}$$

$$diameter(u,v) = \begin{cases} true & length(u,v) = dmr(G) \\ false & otherwise \end{cases} \tag{4}$$

The relation *adjacent* corresponds to the length of an edge in a 3 dimensional space. It has the effect on the regularity of each polygon or face. The relation *neighbour* expresses that the length of path between two vertices is 2, and two vertices are adjacent. It has the effect on the regularity of each vertex figure. The relation *diameter* corresponds to the circum-sphere of polyhedron.

The interaction in the wire-frame subsystem requires the repetition of internalist anticipation and externalist anticipation of the learner [6]. For example, stable wireframe corresponding to a polyhedral graph is not unique in a 3D space (Figure 3).

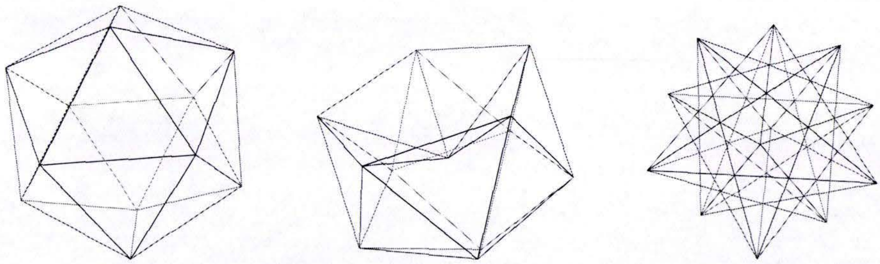

Figure 3: Examples of non-unique stable states corresponding to an icosahedral graph.

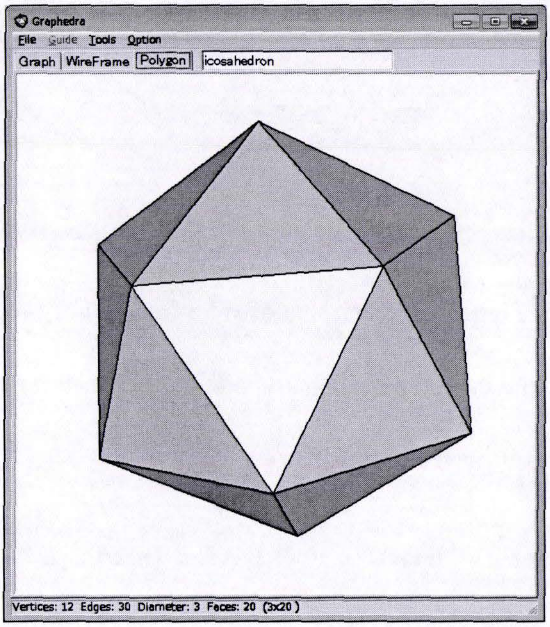

Figure 4: Screen shot of Polygon Subsystem.

2.3 Polygon subsystem

After arranging vertices in 3D space, the last step is detecting faces, selecting appropriate faces, and rendering the solid. Figure 4 shows a screen shot of polygon subsystem. Detecting n-polygon is equivalent to finding simple closed path with length n. Some additional utilities such as opening faces, meshed faces are implemented. The interaction in the polygon subsystem is based on both of internalist anticipation and externalist anticipation (Section 2.5).

2.4 Graph Operation for Polyhedral Graph

Three graph operations are defined for polyhedral graphs: *vertex splitting*, *edge contraction*, and *diagonal addition* (Figure 5) [4]. By these 3 operations, 5 regular polyhedra (Platonic solids) and 13 semi-regular polyhedra (Archimedean solids) are interconnected as shown in Figure 6. The symbol δ stands for applying *diagonal addition* to appropriate quadrangular faces. The operator $[\![m, n]\!]$ stands for applying *edge contraction* to each edge between two faces of F_m and F_n, where F_n indicates the set of faces with n edges. The operator σ stands for applying *vertex splitting* to every vertex. By using these operations, the user can model various anticipated polyhedra from one seed polyhedron.

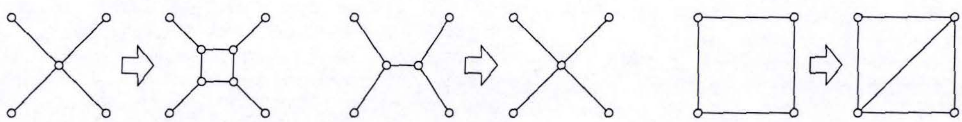

Figure 5: Three Graph Operations.

(a) Vertex splitting, (b) Edge contraction, and (c) Diagonal addition.

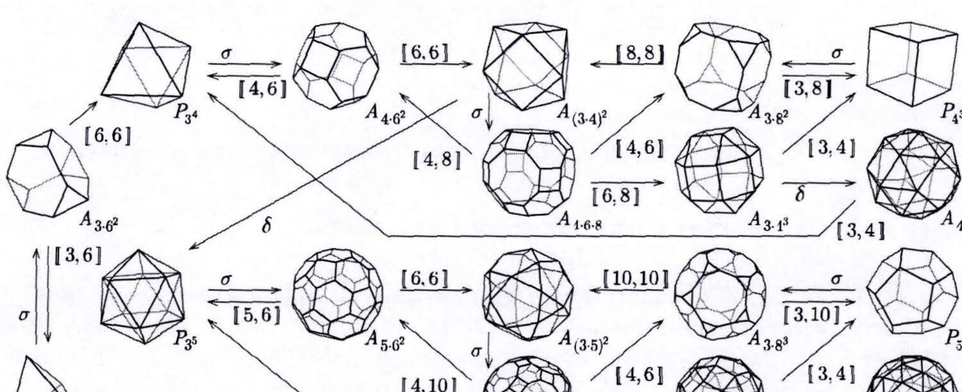

Figure 6: Relations of 5 Platonic graphs and 13 Archimedean graphs using 3 operations.

2.5 Selectability of Vertices Alignments and Polygons.

Wire-frame of icosahedral graph has 2 symmetric stable states in a 3D space. They can be formed by handling virtual elasticity. The upper half of Figure 7 shows the 2 different states corresponding to icosahedral graph. In the lower left of Figure 7, by detecting 20 triangles or 12 pentagons, icosahedron or great dodecahedron is obtained respectively. In the lower right of Figure 7, by detecting 12 pentagrams or 20 triangles, small stellated dodecahedron or great icosahedron is obtained respectively. Because such ambiguities exist, both of internalist and externalist anticipation are required to the learner.

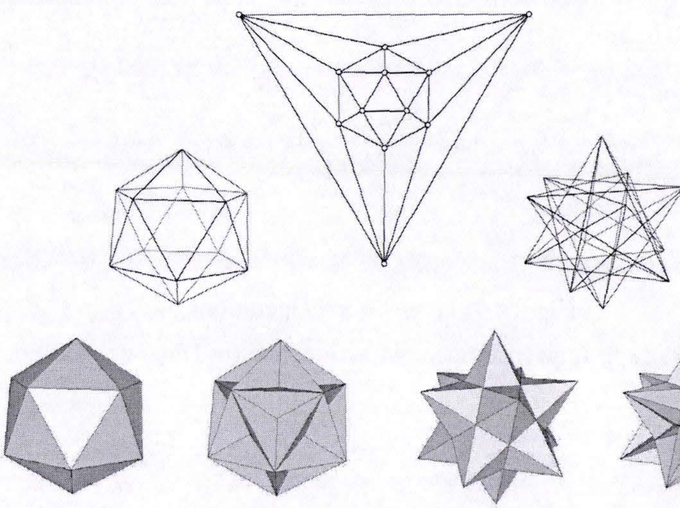

Figure 7: 2 alignments and 4 uniform polyhedra from icosahedral graph.

3 Interactive Vertex Colouring

Vertex colouring of a graph is the assignment of labels to the vertices of the graph so that adjacent vertices have different labels [7-9]. The 4-colour theorem proved by Appel and Haken in 1977, indicates that every planar graph is 4-colourable. Every polyhedral graph is 3-connected planar graph, according to the theorem by Steinitz. Therefore, it is also 4-colourable. Consequently, the chromatic number of a polyhedral graph is 2, 3, or 4. There are various colouring methods, for example, greedy colouring algorithm, sequential colouring algorithm, distributed algorithm, decentralized algorithm, and so on. Determination of 2-colourability is equivalent to testing bipartiteness, therefore, it is computable in linear time. However, in the case of more than 2 colouring, the computational complexity is known to be NP-complete, even for 3-colourability [10], and 4-colourability [11].

A graph G is k-colourable if and only if G is k-partite. In the case of polyhedral graph, k can be 2, 3 or 4. By identifying vertices in each part of k-partite graph, one of line segment (1-simplex), triangle (2-simplex), or tetrahedron (3-simplex) is obtained. In this paper, we call such a polytope generated from a polyhedron, *degenerated* polyhedron.

Figure 8 shows three examples of degenerated polyhedra: snubdodecahedron (to tetrahedron), smallrhombicosidodecahedron (to triangle), and greatrhombicosidodecahedron (to line segment).

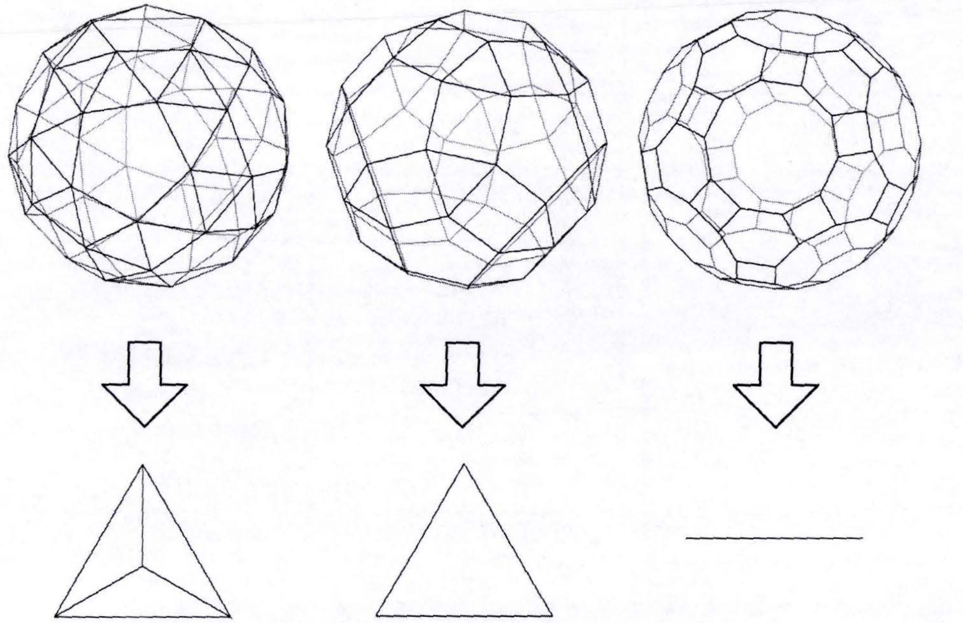

Figure 8: Examples of degenerated polyhedra.

The process of interactive vertex colouring is as follows. When the user presses the "Degenerate" button, the system tests the chromatic number in the order of 2, 3, and 4 colourabilities. If the graph is k-colourable, it is partitioned to k-partite graph, and degenerated to one of line segment, triangle, or tetrahedron, with animations. At this point the learner can anticipate the result of vertex colouring. After the user select different colour for each vertex of degenerated polyhedron, when the user presses the "Release" button, the polyhedron recovers the original shape with also animations. The user or learner can observe how the colours are assigned to the vertices so that adjacent vertices have different colours, and also understand unconsciously that k-colourable and k-partite are equivalent. Figure 9 shows an example of interactive vertex colouring with a time series of screen shots.

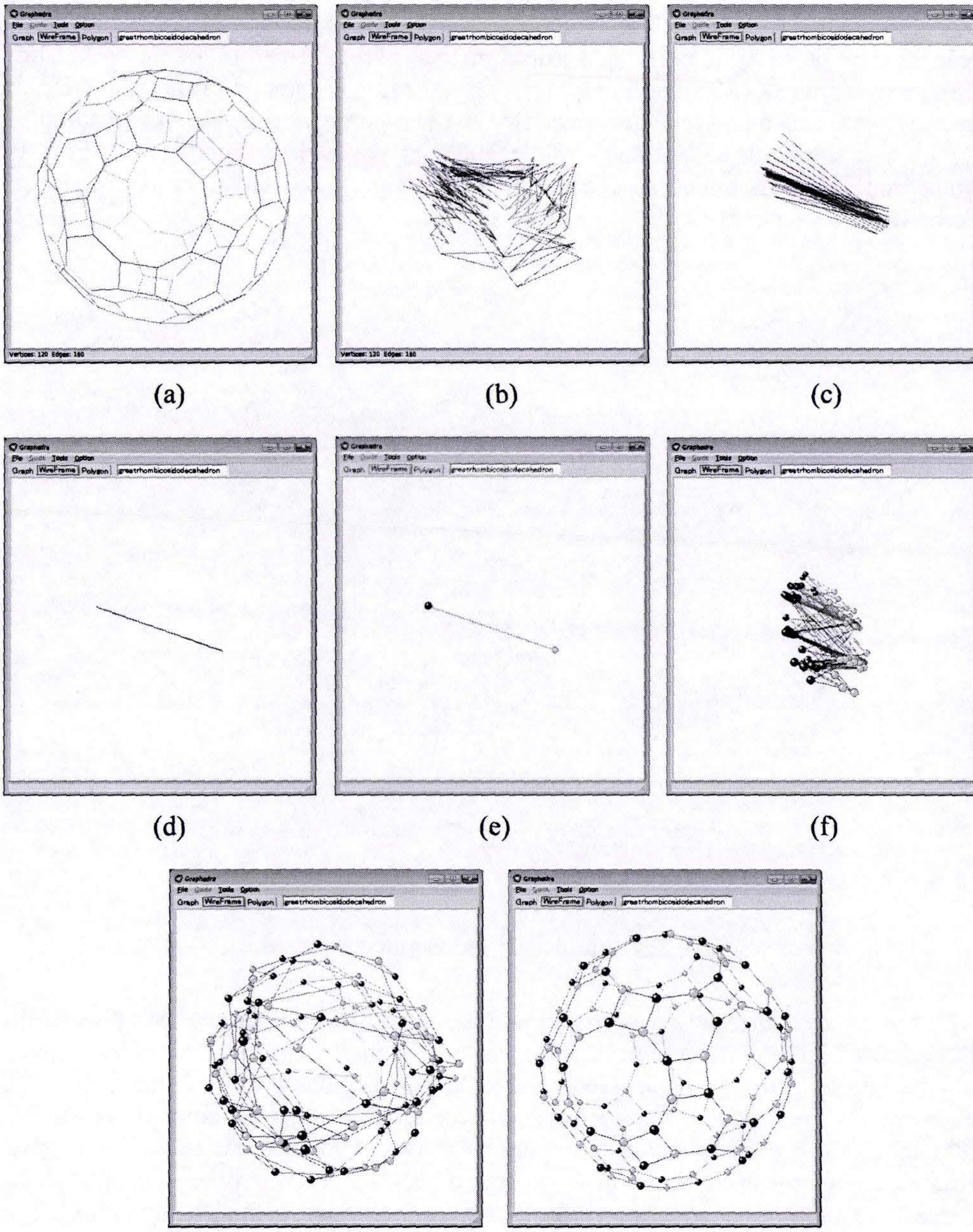

Figure 9: An example of interactive vertex colouring.
In this example, a greatrhombicosidodecahedron is degenerated to a line segment, which means this polyhedral graph is bipartite.

4 Conclusion

In this paper, interactive vertex colouring system for polyhedral graph has been presented. It was developed as a subsystem of an interactive learning system of polyhedra, based on graph theory. The interaction is promoted by iteration of internalist anticipation and externalist anticipation. The learner can not only observe how the colours are assigned to the vertices, but also understand unconsciously that the notions of *k*-colourable and *k*-partite are equivalent.

References

[1] Nonaka, H. (2008) Detection of Faces in Wire-Frame Polyhedra, Proceedings of International Conference on Computer Graphics Theory and Applications, 3, 134-137.

[2] Nonaka, H. (2008) Modeling of Kepler-Poinsot Solid Using Isomorphic Polyhedral Graph, International Journal of Computer Science, 3(1), 64-67.

[3] Nonaka, H. (2008) Interactive Modeling of Uniform Polyhedra by Means of Graph Drawing, Proceedings of International Conference on Information Technology and Applications, 5, 822-827.

[4] Nonaka, H. (2008) Operation-Based Notation for Archimedean Graph, Journal of Systemics, Cybernetics and Informatics, 6(6), 19-22, 2008.

[5] Nonaka, H. (2009) Visualization of Conway Polyhedron Notation, Proceedings of World Congress on Science, Engineering and Technology, 38, 692-695.

[6] Dubois, D.M.(1998) Introduction to Computing Anticipatory Systems, International Journal of Computing Anticipatory Systems, 2, 3-14.

[7] Bondy, J.A. and Murty, U.S.R. (2008) Graph Theory, Springer.

[8] Gross, J. and Yellen, J. (1998) Graph Theory and Its Applications, CRC Press.

[9] Nishizeki, T. and Chiba, N. (1988) Planar Graphs, Dover Publications Inc.

[10] Beigel, R. (2005) 3-Coloring in time $O(1.3289^n)$, Journal of Algorithms 54(2), 168-204.

[11] Byskov, J.M. (2004) Enumerating maximal independent sets with applications to graph colouring, Operations Research Letters, 32, 547-556.