

The Puzzle Layout Problem

Kozo Sugiyama¹, Seok-Hee Hong², and Atsuhiko Maeda³

¹ School of Knowledge Science, Japan Advanced Institute of Science and Technology,
Asahidai 1-1, Tatsunokuchi, Nomi, Ishikawa, 923-1292, Japan
sugi@jaist.ac.jp

² School of Information Technologies, University of Sydney, NSW 2006, Australia
shhong@it.usyd.edu.au

³ NTT Network Innovation Labs, NTT Corporation,
Hikarinooka 1-1, Yokosuka, Kanagawa, 239-0847, Japan
maeda.atsuhiko@lab.ntt.co.jp

Abstract. This poster describes an implementation of the puzzle generators and puzzle layouts introduced in [1].

1 Introduction

We present a new application of graph drawing, puzzle layout. We analyze the operations of the puzzles and derive two abstract models which can be modeled as puzzle graphs. Based on these models, we implement two puzzle generators and produce various layouts of the puzzles using various graph drawing algorithms. Using these puzzle generators, we can parametrically change the levels of difficulty of the puzzles and create new puzzles. Further by applying various graph drawing algorithms, we can create new user interfaces for a puzzle. For details, see [1].

The puzzle layout should be very beautiful and attractive to the users. Further, the layout should operate as a puzzle interactively by the users. We applied various standard graph drawing methods to produce puzzle layouts such as spring embedder algorithm, orthogonal drawing and visibility representations. This poster describes an implementation of the puzzle generators and puzzle layouts presented in [1].

2 The Puzzle Generator and the Layouts of Puzzles

Figure 1 shows a user interface of the puzzle generator and the layout. Using the interface, puzzles can be defined in a text-based dialog box. We use a spring algorithm with manual change to produce the layout.

To operate the layout as a puzzle, the user can simply use drag and drop to move an element or to rotate a cycle. Then the cycle which contains the element and the other cycles which are included in the operation rotate together.

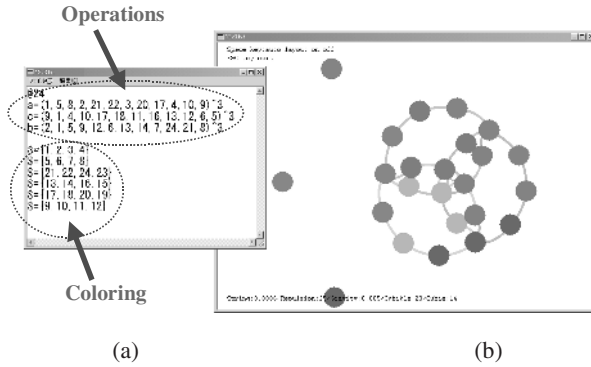


Fig. 1. User interface of a permutation puzzle generator and its layout.

Figure 2 shows three different layouts of the 2^3 Rubik's Cube and the Pyraminx.

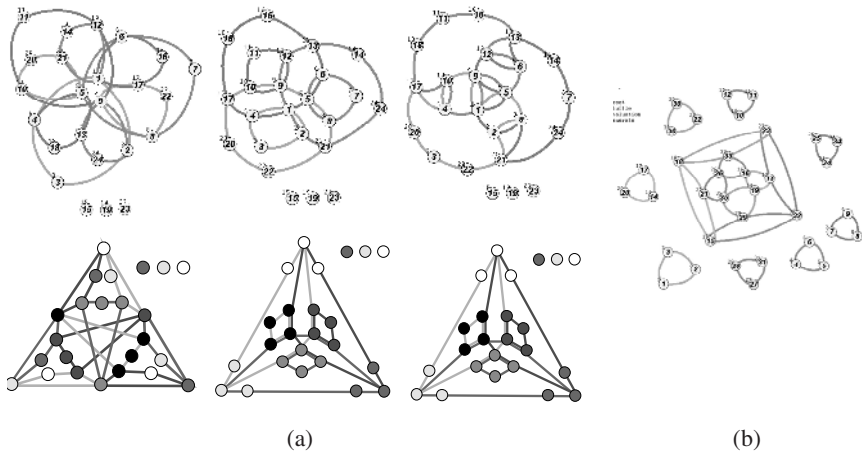


Fig. 2. Layouts of (a) 2^3 Rubik's Cube and (b) Pyraminx.

3 Symmetric Layout Algorithm for Puzzles

We observed that symmetry is the far most important aesthetic criteria in the puzzle layout problem. However, the spring algorithm sometimes fails to achieve a good symmetric layout. Our current work is to implement the symmetric puzzle layout algorithm presented in [1].

References

1. K. Sugiyama, S. Hong and A. Maeda, The Puzzle Conversion and Layout Problem, KS-RR-2003-002, School of Knowledge Science, JAIST, 2003.