Foot Note 828 by Patrick Prosser
Limited Circulation[1] October 16, 2002

A Lesson from the Crystal Maze

Introduction

One of the lovely things about having a young daughter, is that she sometimes forces me to do things that I wouldn't do otherwise, only to discover something as a result (and of course, having a passport to go and see all those U rated movies). Recently Zoe forced me to watch the Crystal Maze, a rather annoying television program. Teams have to perform a number of tasks, mostly physical tasks, rarely intellectual. In this episode one of the tasks was a puzzle. It was in fact a lovely constraint satisfaction problem, realised in large wooden discs with connecting rods. Zoe and I quickly got hold of pen and paper (actually, paper towel) and drew the problem out before we forgot. We also taped it (belt and braces). Unfortunately none of the contestants could solve the problem in the time given (about 3 minutes), but neither could we!

I am soon to teach a 4th year course on constraint programming, and I've been wondering what my first lecture would look like. This is important, because I need to set the scene and get the students interested. I think the puzzle I am about to present is ideal for this purpose, as it allows the students very quickly to get an idea of just what constraint programming is about, and just how much fun can be had.

The Crystal Maze Puzzle

We are given 8 discs, numbered 1 to 8. You must place discs on the nodes of the graph over leaf such that adjacent nodes are not numbered consecutively. You have 5 minutes to do this, starting now!

Representation

How might we represent this problem? We might have a variable for each node, allowed to take a value in the range 1 to 8. For the edges, we have a relation that forces adjacent nodes to take non-consecutive values. In addition, we must introduce a new constraint such that all nodes take different values. Now, number the nodes respecting the constraints, again in 5 minutes!

But wait. Isn't there another representation? We could have 8 variables, where a variable corresponds to a number and the value that variable is assigned corresponds to the node the disc is placed on. We have a symmetrical representation. Could we exploit this in some way? We'll ignore this for the time being but we really should return to it.

Inference

What happens when I place a disc onto a node on the graph? What inferences can I make? Clearly if we place a disc onto a node we can deduce exactly what discs cannot be placed on adjacent nodes. I could demonstrate this to you. And here's a wee question: which node, initially, would result in the greatest amount of inferencing?

[1] I have decided to take my work underground, to stop it falling into the wrong hands.

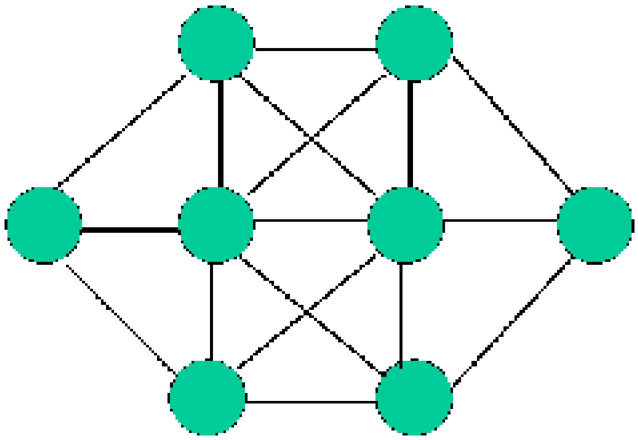

Figure 1: The Crystal Maze Puzzle

Heuristics

What decision do you want to make first? Does it matter? I would suggest that we choose a node that is connected to most others, wouldn't you? Why?

Are all values equal? Are all discs the same? For example is the disc numbered 5 as easy to use as the disc cumbered 8? Disc 5 has disc 4 and disc 6 consecutively below and above it. How about 8? So, would it help to make a hard decison early and get it over with, and by the way, when making that decision try and get away with it?

Conclusion

This is a nice puzzle. It gives the student an intuitive grasp of representation, constraint propagation, variable and value ordering heuristics, and maybe even search. You could draw it on a sheet of paper, copy it, and give each student one to work on[2]. You might care to use it the next time you introduce constraint programming to an audience. But if you do, remember to tell them that it's due to Zoe, Patrick, and the Crystal Maze.

[2]Phil Kilby encouraged me to do this in an algorithms lecture when presenting the travelling salesman problem. We had a sheet with 100 random points on it. The students were to draw a good tour. They had 10 minutes to do this. I then collected the sheets and made comments, and then showed them some really good tours. It went really well.