Constraint Technology for Solving Combinatorial Problems
Project Option 1: Maximum Density Still Life

Magnus Ågren
Computing Science, Information Technology, Uppsala University
agren@it.uu.se

— To be handed in before 08:00 on June 5, 2006 —

Submission Instructions

1. The project report should be handed in individually. Do not work in pairs and hand in nearly identical solutions. It is allowed to discuss solution alternatives but not to copy each other’s programs.

2. Make sure you understand the problem and have an idea about how to solve it before you start programming. Ask for help if you feel that something is unclear.

3. Write clear and understandable code and documentation and state any assumptions you make.

4. Include external documentation describing how your program works. This includes describing the model and any particular relevant features, e.g., heuristics used, search technique(s), special data structures (if any), etc. Explain how to compile and run your program.

5. Include sample test runs of your program with inputs and outputs. Make sure that the test runs are reproducible by the program you hand in.

6. Discuss your results where this is relevant.

7. Submit your solution via the course manager before the deadline. Make a tar or zip archive of all your files. Only include .pdf or .txt files other than your source code files.

Note that failure to follow the instructions above may result in a U grade.
Figure 1: Four states of the Game of Life where \( n = 4 \).

The Game of Life

The following description is inspired by [1]. The *Game of Life* was invented by John Horton Conway in the 1960s. The game is played on an \( n \times n \) board. On this board, each cell is either *dead* or *alive*. A *state* of the game is a board where some of the cells are alive and the rest of the cells are dead. Figure 1 shows four states of the game where \( n = 4 \) and where dead cells are empty while alive cells contain a black checker. Note that each cell on the board may be dead as seen in state (a).

We assume that the board is extended infinitely by dead cells in all directions. Then each cell \( C \) has eight neighbours as seen below:

\[
\begin{array}{ccc}
N1 & N2 & N3 \\
N4 & C & N5 \\
N6 & N7 & N8 \\
\end{array}
\]

The game is played over a number of time periods. Between two time periods \( t \) and \( t + 1 \), the state may change with respect to certain rules, as follows:

- if a cell has exactly three living neighbours at time \( t \), then it is alive at time \( t + 1 \).
- if a cell has exactly two living neighbours at time \( t \), then it is in the same state at time \( t + 1 \) as it was at time \( t \).
- in all other cases, then the cell is dead at time \( t + 1 \).

This is illustrated in Figure 1 above where the states (b) to (d) may be seen as three consecutive time periods. In (b), the dead cell (1, 0) has three living neighbours which implies that it is alive in state (c).\(^1\) Similarly, the dead cell (0, 2) also has three living neighbours and it is alive in (c). The alive cell (0, 1) in (b) only has one living neighbour which implies that it is dead in (c). The same thing holds for the alive cell (1, 3).

A *still life* is a state at some time \( t \) that does not change at time \( t + 1 \). For example, the following state at some time \( t \):

\[
\begin{array}{ccc}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\end{array}
\]

does not change at time \( t + 1 \) according to the rules above (check this). Another example of a still life is the state (a) in Figure 1.

A *maximum density still life* is a still life on an \( n \times n \) board where the number of alive cells is maximised. For example, the still life above is also an example of a maximum density still life for \( n = 3 \). Indeed, there is no other \( 3 \times 3 \) still life where the number of alive cells is greater than six.

---

\(^1\)We use the convention (column, row) and number from zero, so (1, 0) means the cell in the second column and first row.
The Task

Write a constraint program for finding and displaying the first maximum density still life given a positive integer $n$. Your program should at least be able to give the first optimal solution for $n = 8$ within 10 minutes of CPU time on debet.it.uu.se.

Bonus-Points Competition

There is a competition for this project where the first (and only) prize is 15% bonus points on the written exam. To participate in the competition, clearly state that you wish to do so and report the largest $n$ that your program can find the first optimal solution for within 10 minutes of CPU time on debet.it.uu.se. Note that there may be several winners if several people reach the same result.

References