Applied Mathematics and Computation 232 (2014) 292–302

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Minesweeper strategy for one mine

Shahar Golan

Department of Computer Science, Ben-Gurion University of the Negev, POB 653, Beer-Sheva 84105, Israel

ARTICLE INFO

Keywords:
Graph theory
Strategy
Probability
Algorithm
Games
Minesweeper

ABSTRACT

Minesweeper is a popular single player game. Various strategies have been suggested in order to improve the probability of winning the game. In this paper, we present an optimal strategy for playing Minesweeper on a graph when it is known that exactly one cell is mined.

1. Introduction

Minesweeper is a popular single player game, distributed with the Microsoft Windows operating system. It consists of a grid of covered cells (see Fig. 1), some of which are *mined cells*, i.e., contain mines. The cells not containing mines are *free cells*. The mined cells are randomly and uniformly distributed in the grid. Each free cell contains information regarding the number of its mined neighbors (where cells adjacent diagonally are also considered as neighbors). At each move, the player may uncover a cell. If the cell is mined, the game is lost; if it is free, the information contained in it is revealed, and the player may use it. The goal of the game is to uncover all free cells, leaving all mined cells covered (see Fig. 2). The total number of mines is usually given as well.

Various approaches were examined for planning a strategy for maximizing the probability of winning Minesweeper. In [1], software is provided, enabling one to add his strategy (as a Java program) and test its performance on a predefined benchmark. This benchmark consists of three grids: Beginner – an 8×8 grid with 10 mines, Intermediate – 15×13 with 40 mines, and Expert – 30×16 with 99 mines, where the mined cells are chosen in each case uniformly at random out of all possibilities. Several basic strategies are also provided in [1]. One of these regards the problem as a constraint satisfaction problem. A more sophisticated version of this strategy (devised independently) was examined in [2]. The first strategy was found to achieve winning rates of 71%, 36% and 26%, respectively. The second achieved winning rates of 80%, 44% and 34%, respectively for the three grids above. The statistics were taken in a setting where the first move is always safe (i.e., the mines are distributed between all cells except for the one selected by the player at the first move). In [3–5], genetic algorithms were used in order to solve the problem. A learning agent was introduced in [6]. The results of these AI attempts were no better than the human tailored algorithms. There have been also some attempts to use neural networks [7]. Another approach was presented in [8], where the grid is represented by a cellular automaton.

The configuration given to the player before each move, consisting of a grid with some uncovered cells containing known information will be referred to as a *Minesweeper grid* (henceforth MS grid). In such a grid there might be covered cells that can be positively inferred as free. Such cells are *safe*. A rational player may be assumed to always try to find and open a safe cell before guessing. An *assignment* $\mathcal{A}$ is a function from the Minesweeper grid's cells to the set {free, mined}. A *legal assignment* is an assignment such that all uncovered cells are assigned free and contain the correct values (i.e., exactly

E-mail address: golansha@gmail.com

http://dx.doi.org/10.1016/j.amc.2014.01.045
0096-3003/

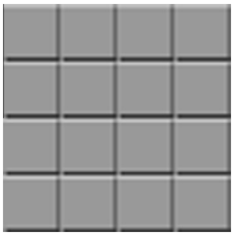
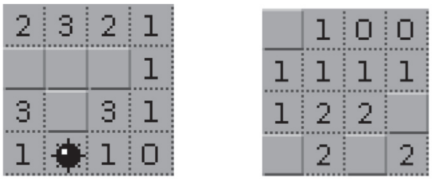


Fig. 1. The initial state.



(a) A lost game (b) A winning state

Fig. 2. Ending states of Minesweeper

the number of mined neighbors they have). A covered cell is safe if and only if all legal assignments give it the value ‘free’. Thus, an important part of any strategy is determining whether there exists a legal assignment where a given cell is mined.

Several variations of the Minesweeper game were defined and analyzed. One of these is to discard the requirement that the board must be a grid, and consider Minesweeper on general graphs. An *MS graph* is the analogue of an MS grid for graph. Thus, in an MS graphs, each free vertex contains the information regarding the number of its mined neighbors. A method for calculating the number of legal assignments for a partially uncovered MS graph is given in [9]. A new version of Minesweeper, where the board consists of triangle cells is presented in [10]. Infinite versions of Minesweeper are given in [11,12].

In this paper we design an optimal strategy for playing Minesweeper on general graphs, when it is known that exactly one cell is mined.

In Section 2 we present the main results. In Section 3 we introduce a concept called *guess component*. The proofs regarding Minesweeper problems with one mine are given in Section 4.

2. The main results

A *strategy* for minesweeper is an iterative algorithm, which determines which cells to uncover in the course of the game. In every iteration of the algorithm, a covered cell is chosen and uncovered. Thus, for any MS graph (not necessarily containing exactly one mined cell), any strategy can be described by the scheme given in Algorithm 1.

Algorithm 1. A generic scheme for a strategy

GenericStrategy (Graph G)
  `Initialize auxiliary data structures`
  `Set` $Covered \leftarrow V(G)$
  while $|Covered| > TotalNumberOfMines$
    $v \leftarrow$ Choose $(G, Covered)$
    `uncover` v
    if $label(v) = *$
      print `lost`
      return
    `Update auxiliary data structures`
    $Covered \leftarrow Covered \setminus \{v\}$
  print won

For a strategy s and a graph G, denote by $P_s(G)$ the probability of finding all the mines in G without uncovering any of their cells when using s. An *optimal strategy* s^* is a strategy such that $P_{s^*}(G) \geqslant P_s(G)$ for every strategy s. Denote $P^*(G) = P_{s^*}(G)$. In this section we present an optimal strategy for playing Minesweeper on an MS graph known to include exactly one mined cell.