TeddySweeper: A Minesweeper Solver

Tai-Yen Wu[1,a)] Ching-Nung Lin[1,b)] Shi-Jim Yen[1,c)]
Jr-Chang Chen[2,d)]

Abstract: Minesweeper is a Partially Observable Markov Decision Process(POMDP) problem. How to gain more information from current situation affects future decision making significantly. This paper proposes a heuristic solver which outperforms all present minesweeper solvers including pure heuristic and hybrid single player MCTS with heuristic approaches. With our methodology, more hidden information can be extracted from the same situation than previous methods. Furthermore, this solver is practical with efficient performance.

1. Introduction

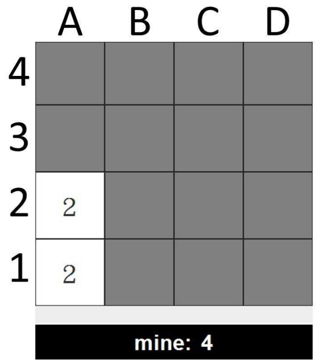

Fig. 1 Single Point Strategy

Minesweeper is not only a NP-Hard[1] problem but also a partial information game. The hidden information problems are occurred in real life all the time. For example, Poker has hidden states. Intuitively, solving a minesweeper game is straightforward. First, (See Figure 1.) Single Point Strategy[2] focuses on *A1* configuration. Because of the rule constraints, *B2* and *B1* are convinced mines. Second, (See Figure 2.) Equation Strategy[2] is more advanced. With some assumption, *C3* and *D3* can be judged as mines. Above two are common strategies for human to solve minesweeper problems.

[1] Dept. of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan
[2] Dept. of Applied Mathematics, Chung Yuan Christian University, Taoyuan, Taiwan
[a)] kingted77513@gmail.com
[b)] jironglin@gmail.com
[c)] sjyen@mail.ndhu.edu.tw
[d)] jcchen@cycu.edu.tw

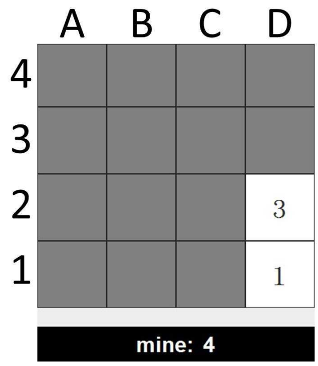

Fig. 2 Equation Strategy

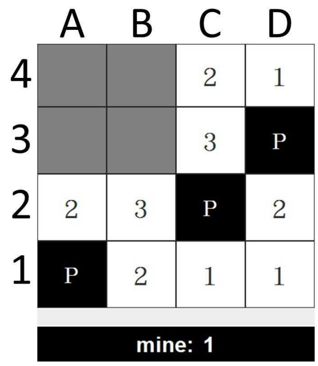

Fig. 3 Constraint Satisfaction Problem

In addition, a computer solver can achieve more. Third, Constraint Satisfaction Problem(CSP)[3] is to list all possible mine combinations to converge a certain location is a mine or not a mine. In Figure 3, there are three combinations as {F,T,F,F}, {F,F,T,F} and {F,F,F,T} representing position *A4*, *B4*, *A3*, *B3* with *F* is equal to "Not Mine" and *T* is equal to "Mine". This method concludes *A4* is not a mine. But, CSP strategy has a serious drawback. Considering twenty unopened cells, there will be maximal 2^{20} combinations. When numbers of unopened cells are big, it's not solvable because of huge number of combinations. Last, the remaining unopened cells require to be solved by guessing. However, A good strategy to guess more precisely remains an open issue.

2. Method

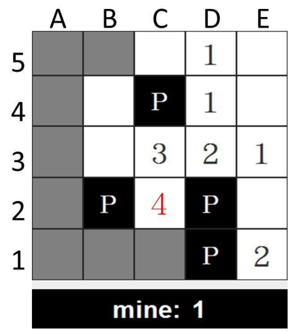

Fig. 4 Calculate the probability if C2 equals to 4

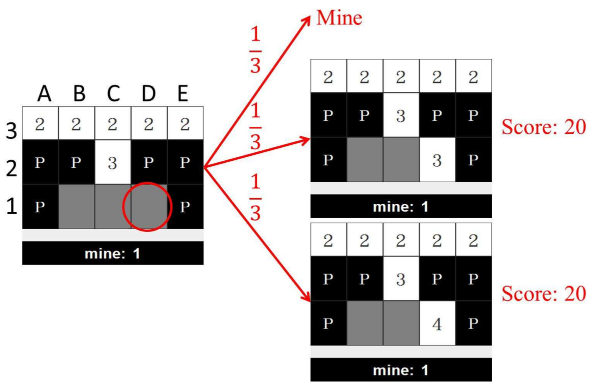

Fig. 5 Calculate all possibilities in D1

Instead of CSP method which considers all combination of current board situation in a game, the proposed method named TeddyMethod calculates further to all possible minefield configuration of each cell and implements a evaluation function to calculate sum of all possible minefield scores. A minefield configuration in a cell is calculated as

$$(1-Mp)*\frac{{}^{bM}P_{needM} * {}^{bL-bM}P_{besideL-needM}}{{}^{bL}P_{besideL}}*\binom{besideL}{needM}$$

where *Neighborhood* (NH) of a cell is legal adjacent cells of this cell. Mp: percentage if this cell is a mine; bM: number of remaining mines in the whole board; $needM$: number of mines required in the unopened cells in the NH of this cell; bL: number of unopened cells in the whole board; $besideL$: number of unopened cells in the NH of this cell.

An example (See Figure 4) is that the cell in position $C2$ has two possible outcomes as number 3 and 4. So, the probability of this cell as a number 4 is $(1-0)*1/8*7/7*\binom{2}{1} = 1/4$. Others can be calculated similarly. A more difficult example (See Figure 6.) is to calculate the percentage of number 3 in position $C2$. The solution differs when C3 or D4 is a mine or not. This can be calculated in the same method with extra combination. (See Figure 7.) Assuming C2 is not a mine, there are six possible situation. The probability of this cell as a number 3 is $((1-0)*1/8*7/7*\binom{3}{1}+(1-0)*1/8*7/7*\binom{3}{1}+(1-0)*2/8*6/7*\binom{3}{2}+(1-0)*1/8*7/7*\binom{3}{1}+(1-0)*1/8*7/7*\binom{3}{1}+(1-0)*2/8*6/7*\binom{3}{2})/6$. As a result, this method can extract more information than any previous method.

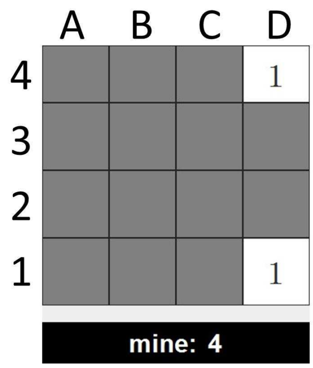

Fig. 6 4 mines in a 4x4 board

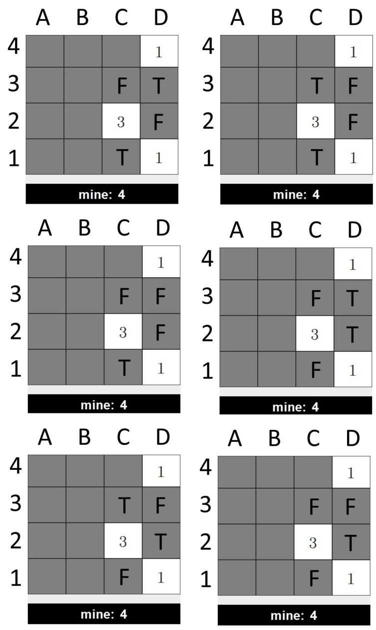

Fig. 7 All variables for this situation

Then, an evaluate function is defined as

$$Score = \sum_{Cellnumber=1}^{8} Case1*20 + Case2*15 + Case3*10$$

. Each Case is Boolean result. *Case1*: an unopened certain position can be searched with Single Point Strategy. *Case2*: an unopened certain position can be discovered with Equation Strategy. *Case3*: no any certain unopened position can be detected. Intuitively, when Case1 happens, the remaining unopened cells are easier to be solved because more information gained. So, Case1 is assigned a higher value. Case2 and Case3 follow the same logic, especially Case3 is set to 10 which is the lowest because uncertain guessing does not provide extra information to solve following unopened cells. For instance, the score of D1 in Figure 5 is 0(№1) + 0(№2) + 1/3 (№3) * 20 + 1/3 (№4) * 20 + 0 (№5) + 0 (№6) + 0 (№7) + 0 (№8) = 40/3. As a result, the score of B1 and D1 are the same which are higher than C1 which score is $2/3*10 = 20/3$. Hence, playing B1 or D1 gains higher solving rate than C1. On the contrary, CSP considers all three positions own equal solving rate as 1/3.

CSP calculates all combination of unopened cells and left mines, which is time consuming. Teddy Method implements a "Divide and Conquer" (D & C) algorithm to reduce the combination dramatically. In some cases, (See Figure8.) D & C algorithm cannot be applied because the two opened cells correlate with each other. Otherwise, in other cases, it can reduce a large number of combinations. (See Figure9 and 10.) With CSP approach, calculating all possible outcomes requires $3*5 = 15$ combination. On the contrary, our algorithm needs $3 + 5 = 8$ combination with a transition function. This transition function is lightweight and maintains result consistency. For example, the result of Area A is scaled to 3 and the result of Area B is scaled to 5. In addition, this can accelerate the speed to calculate all possible minefield configuration which might have enormously combination.

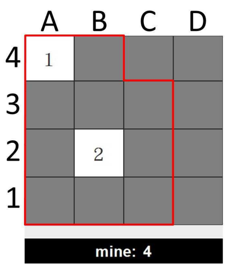

Fig. 8 A situation that Divide and Conquer cannot be applied

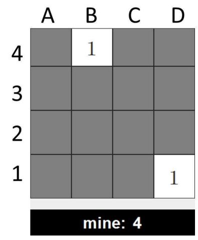

Fig. 9 A situation that Divide and Conquer can be applied

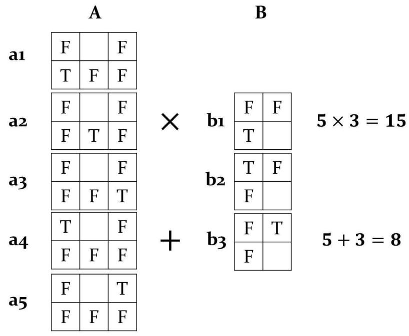

Fig. 10 Divide and Conquer apply on the previous Figure

3. Experimental Result

This solver is implemented in Java with single thread. All experiments are conducted on a Intel Celeron G530 2.4 GHz machine which has 8 gigabyte ram, JDK 1.7 and Windows 7. For 8x8, 9x9 and 16x16 boards, the result are averaged from 100000 games and 10000 games for 16x30 board.

Table 1 and Table 2 show that solving rates of proposed method are significantly better than pure Heuristic HCSP and Hybrid MCTS OH solvers especially in 8x8, 9x9 and 16x16 boards.

Format	HCSP[2]	Proposed Method
10 mines on 8x8	79.9%	81.52% ± 0.25%
10 mines on 9x9	90.5%	91.78% ± 0.17%
40 mines on 16x16	76.4% ± 0.4%	77.84% ± 0.26%
99 mines on 16x30	38.1% ± 0.5%	38.84% ± 0.94%

Table 1 Results (Winning rates) comparing to the strongest heuristic Minesweeper solvers

Format	OH[4]	Proposed Method
10 mines on 8x8	80.2% ± 0.48%	81.52% ± 0.25%
10 mines on 9x9	89.9% ± 0.3%	91.78% ± 0.17%
40 mines on 16x16	74.4% ± 0.5%	77.84% ± 0.26%
99 mines on 16x30	38.7% ± 1.8%	38.84% ± 0.94%

Table 2 Results (Winning rates) comparing to a MCTS Minesweeper solvers

Table 3 describes the whole running time of without and with D & C. The proposed method speeds up dramatically which can overcome CSP drawbacks of slow speed to calculate exponential combination.

Table 4 showed the winning rate in certain time

Format	Without D & C	With D & C
10 mines on 8x8	>1 months	19757s
10 mines on 9x9	>2 weeks	12722s
40 mines on 16x16	>2 months	86150s
99 mines on 16x30	>2 months	57943s

Table 3 Results of total executing time with and without D & C

constraints. The proposed method outperforms traditional CSP because it can compute the result quickly enough instead of sacrificing accuracy limited by time or combination number.

Format	CSP-PGMS	CSP only(TeddyMethod)
10 mines on 8x8	75.90%	78.06% ± 0.28%
10 mines on 9x9	80.00%	89.75% ± 0.23%
40 mines on 16x16	45.00%	74.53% ± 0.92%
99 mines on 16x30	34.00%	36.35% ± 0.94%

Table 4 Comparing CSP with time constraints

4. Conclusions

With such high efficiency method, it archives the present world best minesweeper solver. It is possible to design an automatically adjusting parameter mechanism for the proposed evaluation function to gain higher solving rates. Also, the D & C CSP algorithm can apply on other problems. In the future, combining the proposed method with single player MCTS might increase solving rates since this method outperforms other heuristic methods.

References

[1] Allan Scott, Ulrike Stege, and Iris Van Rooij, "Minesweeper May Not Be NP-Complete but Is Hard Nonetheless", The Mathema tical Intelligencer, 2011, Pages 5-17.

[2] Dmitry Kamenetsky and Choon Hui Teo, "Graphical Models for Minesweeper Project Report", 2007.

[3] Chris Studholme, "Minesweeper as a Constraint Satisfaction Problem", Unpublished project report, 2000.

[4] Olivier Buffet, Chang-Shing Lee, Woan-Tyng Lin, and Olivier Teytaud, "Optimistic Heuristics for MineSweeper", Internatio nal Computer Symposium, 2012, Pages 199-207.