The AAAI-17 Workshop on
What's Next for AI in Games?
WS-17-15

Exploring Efficient Strategies for Minesweeper

Jinzheng Tu
Electrical Engr.
Tsinghua Univ.
Beijing, China

Tianhong Li
IIIS
Tsinghua Univ.
Beijing, China

Shiteng Chen
Inst. of Software
CAS
Beijing, China

Chong Zu
Dept. of Physics
Univ. of California
Berkeley

Zhaoquan Gu
Dept. of CS
HKU
Hong Kong, China

Abstract

Minesweeper is a famous single-player computer game, in which the grid of blocks contains some mines and the player is to uncover (probe) all blocks that do not contain any mines. Many heuristic strategies have been prompted to play the game, but the rate of success is not high. In this paper, we explore efficient strategies for the Minesweeper game. First, we show a counterintuitive result that probing the corner blocks could increase the rate of success. Then, we present a series of heuristic strategies, and the combination of them could lead to better results. We also transplant the optimal procedure on the basis of our proposed methods, and it achieves the highest rate of success. Through extensive simulations, a combination of heuristic strategies, "PSEQ", yields a success rate of $81.627(8)$ %, $78.122(8)$ %, and $39.616(5)$ % for beginner, intermediate, and expert levels respectively, outperforming the state-of-the-art strategies. Moreover, the developed quasi-optimal method, combining the optimal procedure and our heuristic methods, raise the success rate to at least $81.79(2)$ %, $78.22(3)$ %, and $40.06(2)$ % respectively.

1 Introduction

Minesweeper is a popular single-player game which is included in many operation systems, such as Microsoft Windows and Linux. The game consists of a grid of blocks, and some blocks contain hidden "mines". Initially, all blocks remain covered, and the player only knows the grid's width, height, and the number of mines. In each step, the player can uncover or probe a covered block. If the block contains a "mine", the player loses the game; otherwise, the block returns the number of adjacent blocks that contain "mines". If all blocks that do not contain any "mine" have been probed, the player wins the game.

Although the simplicity of the rules makes the Minesweeper game interesting and deceptively easy, the Minesweeper game is theoretically proved to be NP-Complete by Kaye (2000). Furthermore, Pedersen proved Minesweeper assignments to be # P-complete. Given any configuration containing the uncovered and covered blocks, the *Minesweeper Consistency problem*, which is to assign the mines to the covered blocks of the configuration such that the uncovered blocks have the correct number of adjacent mines, is shown to be NP-Complete. Moreover, the Minesweeper Counting problem, which is to count the number of legal and consistent assignments of the configuration, is also #P-complete (Pedersen 2004).

The Minesweeper game has attracted much attention in the last decades, and many strategies have been proposed. The majority of them implement combinations of constraint-based approaches and heuristic methods to achieve better success ratio. The former is to deduce which blocks are free of mines, i.e. the blocks do not contain a "mine" definitely, while the latter is to select which blocks are better for probing when every covered blocks may contain mines. There are two types of heuristic methods: deterministic algorithms result in the same choice to probe when the player faces the same Minesweeper configuration, while *non-deterministic algorithms* could lead to different options for probing. The native method of choosing the blocks with least probability of mine wins only about 35 % of the expert level games (30×16 grid with 99 mines); Studholme (2000) tries to probe the block in the corner, winning 33.9 % of expert level deterministically. Legendre et al. (2012) minimizes the distance towards the boundary, winning 37.5 % of expert level. Buffet et al. (2013) introduces upper confidence trees (UCT, first introduced by Kocsis and Szepesvári) for the heuristic methods, winning $38.7(18)$ % of expert level non-deterministically.

In this paper, we are to explore efficient algorithms for increasing the rate of success. In traditional Minesweeper rules, the first step of probing is certainly safe, which may play an important role in designing efficient algorithms. In this paper, we firstly show that we can improve the survival probability within two steps by 4.4 % by probing the corner in the first step. Secondly, we introduce several heuristic methods for handling the game, of which combinations could improve the success ratio. Thirdly, we propose quasi-optimal algorithms to enumerate the possible assignments of the configurations when the time complexity is acceptable. Finally, we conduct extensive simulations to evaluate these algorithms, and the results show that our algorithms outperform the state-of-the-art results.

The remaining of the paper is organized as follows. The next section highlights some related works about the Minesweeper game. The preliminaries are presented in Section 3. The impact of the first probe is discussed in Section 4, and heuristic methods handling the game are introduced in Section 5. Section 6 presents the quasi-optimal methods that could achieve better results. We conduct extensive simulations to evaluate the algorithms, and the results are shown in Section 7. Finally, we conclude the paper in Section 8.

2 Related Works

The studies of strategies of Minesweeper date back to 1990s. Despite unique approaches such as cellular automation (Adamatzky 1997), learning algorithms (Castillo and Wrobel 2003; Pena 2004), graphical algorithms (Kamenetsky and Teo 2007; Golan 2011; 2014), and non-deterministic upper confidence trees (Buffet et al. 2013; Sebag and Teytaud 2012a; 2012b), most of them implemented combinations of constraint-based approaches and deterministic heuristic methods.

For the constraint part, the full-scale constraint solver, used by Studholme (2000) and Legendre et al. (2012), is the most popular one. It takes all constraints into consideration at the same time and solve for all possible solutions. However, simplified solvers are also used (Bayer, Snyder, and Choueiry 2006; Pedersen 2004; Collet 2005), considering only locally related constraints. Legendre et al. (2012) demonstrate that the full-scale solver outperform simplified ones regarding the success ratio.

For the heuristic methods, various have been promoted and/or tested. Studholme (2000) developed a heuristic method that prefers probing corners and edges every time. Legendre et al. (2012) tried to maximize or minimize the distance towards the boundary / boundary or frame. Legendre et al. also combined these distance-related methods together and tested these compounded methods' performance.

For the initial move, Buffet et al. (2013) preferred the corner blocks without explanation. Studholme (2000) and Becerra (2015), however, both asserted that the corner blocks are optimal because corner blocks have the highest probability of getting a zero.

3 Preliminary

Suppose the Minesweeper game consists of an $w \times h$ grid, and there are m blocks in the grid containing a "mine". We use the notation "*w*-*h*-T*m*" to represent such configuration. Define G as the grid of blocks, and all blocks are initially covered with no information apprised. In each step, the player can uncover or probe a block to see if it contains a mine or not. Denote C as the set of closed blocks, i.e. covered blocks, and $B = G \setminus C$ as the set of open blocks, i.e. probed blocks. Denote M as the set of blocks containing mines. The player loses the game if any block $b \in M$ is chosen to probe. Define function $U : G \to \mathcal{P}(G)$ where $U(b)$ represents the set of neighboring blocks around block $b \in G$. $\mathcal{P}(G)$ represents the set of G's subsets. Define function $f : B \to \mathbb{N}$ where $f(b) = \|M \cap U(b)\|$ represents the number of mines around block b. Figure 1 gives an intuitive illustration of these notations.

The rule of playing the Minesweeper game may vary by different versions; however, we consider the following rules that are commonly adopted.

1. The first choice of probing a block is guaranteed to be safe; mines are distributed uniformly over all blocks except the initially probed one.
2. The difficulty levels are 8-8-T10 (beginner), 16-16-T40 (intermediate) and 30-16-T99 (expert).
3. The player is allowed to probe any closed blocks, but probing open blocks are forbidden.
4. When a block without mine $b \notin M$ is probed, it is marked as open ($\in B, \notin C$) and the player is informed of $f(b)$. The neighboring blocks $U(b)$ will *not* be probed automatically even if $f(b) = 0$. *In some version of the game, the neighboring blocks will be probed automatically when* $f(b) = 0$. *However, any player with basic knowledge of the game knows that* $U(b) \cap M = \emptyset$ *and then probe the neighboring blocks.*
5. When a block with mine $b \in M$ is probed, the player loses the game immediately.
6. When there are only $\|C\| = m$ closed blocks left, the player wins.

After the first probe, the player could decide which block is safe or have to guess from the closed blocks. Formally, the player should choose a block $b \in C$ to probe with knowledge of U, C, B and $f(b), \forall b \in B$. When all blocks that do not contain mines are uncovered (probed), the player wins the game.

There are two common metrics for evaluating the Minesweeper strategies: average time consumption, representing the time for winning the game; and the rate of success, representing the success ratio when the player or strategy has enough time to make decisions. In this paper, we mainly focus on the rate of success and to design efficient algorithms with high ratios of winning the game with acceptable time complexity. Naturally, the first choice of prob-

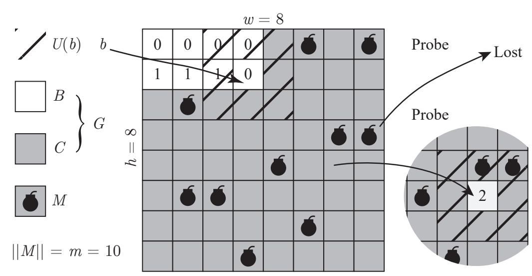


Figure 1: Notation used in this paper and possible response after probing a block.

ing should be discussed separately because of its high symmetricity and simplicity. Subsequently, with blocks gradually uncovered, it is very likely that some blocks can be asserted free of mine. For example, Figure 1 shows an example that the block to the left has $f(b) = 0$, and their neighbors must be free of mines, of which security is logically asserted. However, when all left closed blocks may contain mines, the player has to GUESS with the known configuration, including U, C, B, and $f(b), b \in B$. We formally describe the strategy of handling the game as in Algorithm 1.

Algorithm 1 A Generalized Minesweeper Strategy

1: **procedure** STRATEGY(U, C, B, f)
2:     **if** $B = \emptyset$ **then**
3:         Probe the block at some specific position
4:     **else**
5:         $S \leftarrow \{s | s \in \mathcal{P}(C), \|s\| = m\}$
6:         **for all** $b \in B$ **do**
7:             $S \leftarrow \{s | s \in S, \|s \cap U(b)\| = f(b)\}$
8:         $X \leftarrow C \setminus (\cup S)$
9:         **if** $X = \emptyset$ **then**
10:             $X \leftarrow$ GUESS(C, B, f, S)
11:         Probe an arbitrary block $\in X$

The "specific position" in line 3 is where we should choose for the first probe. The function GUESS is used to guess which blocks are better when no blocks are logically free of mine. In the following sections, we will discuss which block is the optimal choice for the first probe, and how to design efficient GUESS functions.

4 Impact of the Initial Probe

There are three types of blocks: *corner blocks* which has only three neighbors, *edge blocks* which has five neighbors, and the other, *central blocks* have eight neighbors. Since the first probe is always safe, the player could get different information by probing different types of blocks. An intuitive view would be choosing a block in or near the middle, which may increase the success ratio since it contains more information than other blocks. However, in this section, we show that probing the corner blocks in the first step could result in higher rate of success in all the three difficulty levels.

Conjecture 1. *Probing the corner blocks at the first move maximizes the rate of success.*

Before we discuss Conjecture 1, we analyze a weak version that the player survives in two moves as Lemma 1.

Lemma 1. *Probing the corner blocks at the first move maximizes the probability of survival within two moves.*

Proof. Assume there are $N = \|U(b)\|$ blocks adjacent to a given one (i.e., three for a corner block, five for an edge block, and eight for central blocks). Denote random variable $X = f(b)$. We can easily deduce that X follows the hypergeometric distribution $(N, w \times h - 1, m)$. Whichever

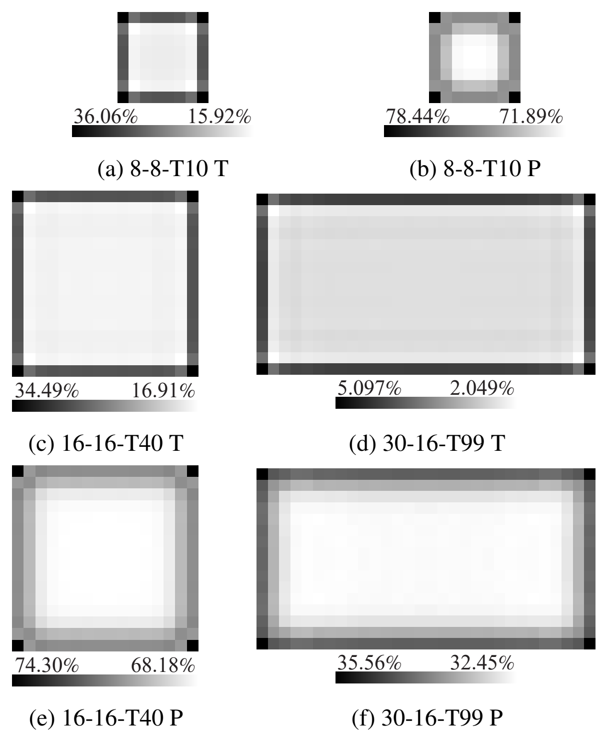
Table 1: Upper-bounds on the success rate on various difficulty levels and initial probing positions.

Difficulty	Corner	Edge	Central
8-8-T10	0.9396	0.9117	0.8983
16-16-T40	0.9381	0.9108	0.8963
30-16-T99	0.8970	0.8614	0.8530
N	3	5	8

strategy we use, the probability of survival following the *second* move is precisely given by:

$$E\left(1 - \min\left\{\frac{X}{N}, \frac{m - X}{w \times h - 1 - N}\right\}\right) \quad (1)$$

Numerically, we evaluated probability in (1) as Table 1. The calculation indicates that an upper-bound on the success rate of any corner-first strategy is higher than that of edge-first or center-first strategies. In other words, not probing the corner first increases the probability of immediate death in the second move significantly. ☐



(a) 8-8-T10 T (b) 8-8-T10 P (c) 16-16-T40 T (d) 30-16-T99 T (e) 16-16-T40 P (f) 30-16-T99 P

Figure 2: Rate of success with different initial position specification on various size of grids. Darker block indicates a higher rate of success. "P" represents the naive guessing, and "T" represents the trivial guessing.

Unfortunately, generalizing Lemma 1 to Conjecture 1 is tough because the number of possible situations balloons rapidly when we consider more moves. However, Monte

Table 2: Impact of the initial probe on the rate of success for various difficulties and guessing strategies.

Config.[1]	Corners[2]	Non-corners[2] Max.	Non-corners[2] Min.
8-8-T10 P	78.44(2) %	74.90(2) %	71.89(2) %
16-16-T40 P	74.30(2) %	71.06(2) %	68.18(2) %
30-16-T99 P	35.56(1) %	34.50(2) %	32.45(2) %
8-8-T10 T	36.06(2) %	29.52(2) %	15.92(1) %
16-16-T40 T	34.49(2) %	28.80(2) %	16.91(1) %
30-16-T99 T	5.097(6) %	4.369(8) %	2.049(6) %

[1] "P" represents the naive guessing, and "T" represents the trivial guessing.
[2] Significant figures are estimated by 0.05 quantiles of β-distribution.

Carlo simulations can be used to consolidate it. We implemented them for each possible position using two different strategies on the three difficulty levels. The first strategy, *naive guessing*, solely uses the naive guessing algorithm by picking the block with least probability of mine. The second strategy, *trivial guessing*, surrenders when it is forced to guess. These two strategies were chosen for their excellent representativeness and simplicity. Each position has been tested 2.5×10^7 times for one combination of difficulty level and strategy, while the corners have been tested 5.0×10^7 times each. Results are partially listed in Table 2, with intuitive figures (2a to 2f) provided as reference. These results show that probing the corner blocks increases the lower bound of the rate of success for whichever strategies or difficulty levels.

Remark. Although further theoretical analysis and numerical experiments are necessary for optimality, Conjecture 1, that choosing corner blocks at the first move is not a bad choice.

5 Heuristic Strategies

With the initial move fixed at the corner blocks, we discuss the heuristic strategies for the Minesweeper game rigorously in this section. The naive heuristic strategy is to minimize the probability of uncovering a mine for a given configuration, and we called the method FILTER-P. Rigorously, let

$$p(b) = P(b \in M) = \frac{\|\{s | s \in S, b \in s\}\|}{\|S\|}$$

and FILTER-P returns the set of $b \in C$ with minimal $p(b)$. It can be thought as maximizing the probability of opening at least one more block (i.e., with minimum probability of uncovering a mine within one step).

We present another method called FILTER-Q, which max-

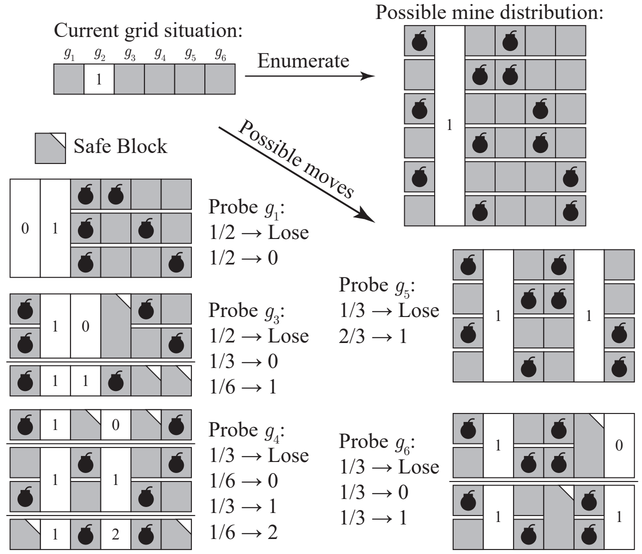


Figure 3: Calculation of $s(b)$ and $e(b)$. $s(b)$ is the probability of obtaining a safe blocks, while $e(b)$ is the expectation of the number of safe blocks. Table 3 gives the result of $s(b)$ and $e(b)$.

imizes the entropy, $q(b)$, of the distribution of $f(b)$:

$$S(b,n) = \{s | s \in S, b \notin s, U(b) \cap s = n\}$$

$$p(b,n) = P(f(b) = n) = \frac{\|S(b,n)\|}{\|S\|}$$

$$q(b) = -\sum_{i=0, p(b,i)\neq 0}^{\infty} p(b,i) \log_2 p(b,i)$$

We introduce two more strategies that are related to the safe blocks, i.e. the blocks are guaranteed without any mines. FILTER-E maximizes the expected number $e(b)$ of safe blocks for the next move, and FILTER-S maximizes the probability $s(b)$ of obtaining at least one safe block for the next move. The $\chi : \mathbb{N} \to \mathbb{R}, x \mapsto \chi(x)$ is the function that returns 0 when $x = 0$ and returns 1 when $x \geq 1$.

$$e(b) = \sum_{n=0}^{\infty} p(b,n) \|C - \{b\} - (\cup S(b,n))\|$$

$$s(b) = \sum_{n=0}^{\infty} p(b,n) \chi(\|C - \{b\} - (\cup S(b,n))\|)$$

Arising from Lemma 1, we introduce FILTER-U, which maximizes $u(b)$, the probability of survival after two steps.

$$p'(b,n) = 1 - \min_{c \in C - \{b\}} \frac{\|\{s | s \in S(b,n), b \in s\}\|}{\|S(b,n)\|}$$

$$u(b) = \sum_{n=0}^{\infty} p(b,n) p'(b,n)$$

To summarize, we provide intuitive demonstrations of various strategies in a 6-1-T2 Minesweeper game in Figure 3 and Table 3.

Table 3: Examples of heuristic strategies when applied to a 6-1-T2 Minesweeper game with $C = \{g_1, g_3, g_4, g_5, g_6\}$, $B = \{g_2\}$, $f(g_2) = 1$. Blocks preferred by each strategy separately are underlined.

b	g_1	g_3	g_4	g_5	g_6
$p(b)$	$1/2$	$1/2$	$\underline{1/3}$	$\underline{1/3}$	$\underline{1/3}$
$q(b)$	0	0.637	$\underline{1.040}$	0	0.693
$s(b)$	0	$\underline{1}$	$1/2$	0	$\underline{1}$
$e(b)$	0	$\underline{4/3}$	$1/2$	0	1
$u(b)$	$1/3$	$1/2$	$1/2$	$1/3$	$\underline{2/3}$

Actually, the proposed strategies can be combined for making a good choice for probing. For example, we combine FILTER-P, FILTER-S, FILTER-E, and FILTER-Q together, as described in Algorithm 2. We denote such a greedy heuristic algorithm as "PSEQ" (the sequence matters!). Referring to Table 3, we can find that FILTER-P returns $X = \{g_4, g_5, g_6\}$, and FILTER-S reduced it to $X = \{g_6\}$. Obviously, g_3 is not there—it has been eliminated by FILTER-P ahead! Finally, "PSEQ" will choose $X = \{g_6\}$ to probe, which is the same as FILTER-U. Indeed, this combined strategy would work better than any single strategy (including FILTER-U), and we will present the comparison results in Section 7.

Algorithm 2 The compounded heuristic strategy "PSEQ"

procedure HEURISTIC-PSEQ(U, C, B, f, S)
    $X \leftarrow C$
    $X \leftarrow$ FILTER-P(U, C, B, f, S, X)
    $X \leftarrow$ FILTER-S(U, C, B, f, S, X)
    $X \leftarrow$ FILTER-E(U, C, B, f, S, X)
    $X \leftarrow$ FILTER-Q(U, C, B, f, S, X)
    return X

6 Quasi-optimal Strategies

The optimal minesweeper strategy has been discussed by Golan (2014). With $\tilde{B} = B \cup \{b\}$ and $\tilde{f} : \tilde{B} \to \mathbb{N}, \tilde{f}(b) = i, \tilde{f}(x) = f(x)$ for $x \in B$, we define

$$p_{opt}(B, f, b) = \sum_{i=0}^{\infty} p(b, i) p_{opt}(\tilde{B}, \tilde{f})$$

$$p_{opt}(B, f) = \begin{cases} 1 & \|G \setminus B\| = m \\ \max_{b \in B} p_{opt}(B, f, b), & \text{otherwise.} \end{cases}$$

The optimal strategy is to choose the block b with maximum $p_{opt}(B, f, b)$. We denote the strategy as OPTIMAL. Unfortunately, its application is significantly retarded by the time complexity, where the number of function calls it involves to solve even a beginner Minesweeper game has a lower bound of roughly 10^{19}. In this section, we are to combine the optimal procedure and our proposed heuristic strategies for achieving high rate of success regarding for heuristic methods, and decreasing the time complexity regarding for the optimal method.

The intuitive idea is that: the amount of computation becomes feasible in the final phases of the whole game. In such cases, the best choice of blocks can be easily determined by running the OPTIMAL procedure. For the other situations where the recursion is too deep and/or wide, our proposed compounded heuristic method could be applied.

We use the number of situations $\|S\|$ to determine whether OPTIMAL procedure or HEURISTIC strategies should be adopted, as shown in Algorithm 3. By fixing $\|S\|$, we can get a better result with controlled time consumption. Denote a heuristic strategy composed of "P" and the quasi-optimal solver as "P-D*", where "D*" stands for the criterion of $\|S\|$. By combing the heuristic strategies and the optimal procedures, we can improve the rate of success significantly. The simulation results are shown in the next section.

Algorithm 3 The quasi-optimal method "PSEQ-D*".

procedure QUASIOPTIMAL-PSEQ(U, C, B, f, S)
    if $\|S\| \leq$ criterion **then**
        return OPTIMAL(U, C, B, f, S)
    else
        return HEURISTIC-PSEQ(U, C, B, f, S)

7 Simulation

In this section, we conduct extensive simulations to evaluate the performance of our proposed (compounded) heuristic strategies.

To examine which combination of these methods attains maximum performance, we perform Monte Carlo simulation on expert level Minesweeper games, and the results are shown in Table 4, where both the rate of success and the consumed time evaluated. Note that the estimation of time is independent of the estimation of rate of success. For estimating time, the testing platform is Intel®Core™i7-4650U, the sample volume is 10^4, and $5\,\%$ outliers are removed. For estimating rate of success, sample volumes are listed in Table 4, and significant figures are estimated by 0.05 quantiles of β-distribution. Although "U" (short for FILTER-U, the other notations are denoted similarly) significantly outperforms "P" regarding the rate of success, its extended time consumption limited its application. From Table 4, conclusively, "PSEQ" has the best performance regarding the rate of success ($39.616(5)\,\%$) among these combinations.

Similarly, we use numeric simulation to test the quasi-optimal methods' performance. Figure 4 shows the success rates and time complexity using different criterions. It should then be stressed that the quasi-optimal methods made a significant difference as the value of the criterion (x-axis) varies. The rate of success increases as the value of the criterion increases. Nevertheless, the rapidity of the growth in time cost could not be ignored. Different heuristic algorithms (we only plot "P" and "PSEQ" as the representative examples) shared a similar pattern (the trend of increasing success ratio and the time complexity) using quasi-optimal

Table 4: Performance of heuristic strategies.

Strategy	Trials	Rate	Time/ms
U	1.02×10^6	$39.14(9)\,\%$	11.845
P	5.00×10^7	$35.56(1)\,\%$	1.150
S	1.02×10^6	$35.05(9)\,\%$	6.165
E	1.02×10^6	$30.72(9)\,\%$	5.017
Q	1.02×10^6	$13.24(7)\,\%$	1.976
PS	1.00×10^7	$39.25(3)\,\%$	2.191
PU	1.00×10^7	$39.12(3)\,\%$	2.119
PE	1.00×10^6	$38.0(1)\,\%$	2.063
PQ	1.00×10^6	$34.95(9)\,\%$	2.615
PSE	3.00×10^7	$39.60(2)\,\%$	1.831
PSU	3.00×10^7	$39.25(2)\,\%$	1.907
PSQ	5.00×10^6	$39.34(4)\,\%$	2.054
PSEQ	3.20×10^8	$39.616(5)\,\%$	2.153
PSEQU	3.20×10^8	$39.616(5)\,\%$	2.088
PSEUQ	1.00×10^8	$39.61(1)\,\%$	2.101
PSEU	2.00×10^7	$39.57(2)\,\%$	2.059

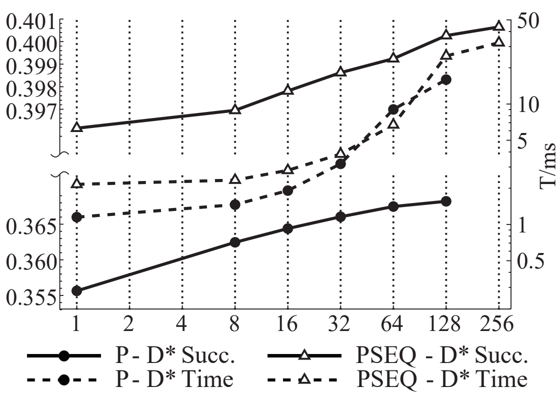


Figure 4: Rate of success and time consumption of quasi-optimal methods.

methods. In addition, from the figure, the quasi-optimal algorithm consumes much more time than traditional heuristic algorithms, where "P-D1" and "PSEQ-D1" represent the traditional "P" and "PSEQ" strategies.

Table 5 lists a comparison of the state-of-the-art deterministic and non-deterministic strategies. The results show that our proposed algorithms have better performance on all three difficulty levels. For the beginner (intermediate, expert) level, the "PSEQ" strategy achieves $1.427\,\%$ ($3.722\,\%$, $0.916\,\%$) higher success ratio than the state-of-the-art algorithms. Moreover, when we adopt the quasi-optimal version, "PSEQ-D256" strategy, the success ratio is $1.59\,\%$ ($3.82\,\%$, $1.36\,\%$) higher than existant algorithms.

Optimistic Heuristics (OH) (Buffet et al. 2013) is the latest result with highest success ratio among the existant works, and our proposed algorithms have better performance than it. We should mention that there exist a slight connection and some clear differences between deterministic quasi-optimal strategies "PSEQ-D*" and non-deterministic strategies OH. The connection is that "looking ahead" is implemented in both. But there are four main differences. First, "PSEQ-D*" postpones the procedure until the final phase and exhausts all possible descendants, whereas OH starts the procedure immediately and stochastically exhausts a certain number of descendants. Second, "PSEQ-D*" enumerates all mine assignments and considers all legal moves, whereas OH expands the upper confidence tree gradually and eliminates some potential moves by some heuristic methods. Third, the set of preferred blocks derived by "PSEQ-D*" under certain situations is always the same as a deterministic strategy, whereas that for OH is likely to vary because of the Monte Carlo part. Finally, the coupling between heuristic methods and "PSEQ-D*" is much looser than that between these strategies and OH, because we can extend any algorithm "X" to "X-D*" easily, whereas OH uses the heuristic ideas to achieve such good result. In general, "PSEQ-D*" boasts more advantages than OH.

From these simulation results, our proposed (compounded) heuristic algorithms could achieve good performance, especially the "PSEQ" method works better than all existant algorithms. Moreover, the proposed quasi-optimal algorithm, which consumes more time than traditional heuristic methods, could improve the rate of success ratio significantly for all three common difficulties: beginner, intermediate and expert levels.

8 Conclusion

In this paper, we study the interesting Minesweeper game by designing efficient algorithms to guarantee a higher rate of success. In the first step, we show a counterintuitive result that probing a corner block at the first move could result in higher rate of success. Then, we propose a series of heuristic strategies, of which combinations improve the rate of success significantly. Following that, we design quasi-optimal algorithms by modifying the optimal procedure with proposed methods. Through extensive simulation results, our proposed deterministic heuristic strategy, "PSEQ", could solve $81.627(8)\,\%$, $78.122(8)\,\%$, and $39.616(5)\,\%$ of beginner, intermediate, and expert levels respectively, outperforming all previous strategies. Moreover, the quasi-optimal methods that boosts the rate of success of "PSEQ" to at least $81.79(2)\,\%$, $78.22(3)\,\%$, and $40.06(2)\,\%$ respectively.

In the future, we are to tighten the upper bound of success rates by the concept introduced in Equation (1), and develop rigorous proof (or disproof) of the conjecture to probing the corner block at the first move. Besides, we could design better heuristic methods and accelerate the quasi-optimal algorithms to improve the rate of success further.

References

Adamatzky, A. 1997. How cellular automaton plays minesweeper. *Applied mathematics and computation* 85(2):127–137.

Bayer, K. M.; Snyder, J.; and Choueiry, B. Y. 2006. An interactive constraint-based approach to minesweeper. In *AAAI*, 1933–1934.

Table 5: Performance comparison of the state-of-the-art strategies and our strategies. The highest rate of success of each difficulty level are marked with bold font.

Strategy	Deterministic	8-8-T10	16-16-T40	30-16-T99
PSEQ-D256 **(this paper)**	Yes	$\mathbf{81.79(2)}\,\%$[1]	$\mathbf{78.22(3)}\,\%$[1]	$\mathbf{40.06(2)}\,\%$[2]
PSEQ **(this paper)**	Yes	81.627(8) %[1]	78.122(8) %[1]	39.616(5) %[2]
OH (Buffet et al. 2013)	No	80.2 %	74.4 %	38.7 %
cSimEnuLoClf (Legendre et al. 2012)	Yes	80 %	75.6 %	37.5 %
CSP (Studholme 2000)	Yes	80.0 %	44.3 %	33.9 %
CSCSP (Becerra 2015)	Yes		75.94 %	32.90 %
LSWPE (Pedersen 2004)	Yes		67.7 %	25.0 %
LBP-MC (Kamenetsky and Teo 2007)	Yes	78.6 %	44.8 %	
PGMS (Ramsdell 1995)	Yes	71 %	36 %	26 %

[1] The sample volume for "PSEQ-D256" is 10^7, and that for "PSEQ" is 10^8.
[2] The sample volume for "PSEQ-D256" is 1.88×10^7, and that for "PSEQ" is 3.2×10^8.

Becerra, D. J. 2015. *Algorithmic Approaches to Playing Minesweeper*. Bachelor's thesis, Harvard College.

Buffet, O.; Lee, C.-S.; Lin, W.-T.; and Teytaud, O. 2013. Optimistic heuristics for minesweeper. In Chang, R.-S.; Jain, L. C.; and Peng, S.-L., eds., *Advances in Intelligent Systems and Applications*, volume 20. Springer Berlin Heidelberg. 199–207.

Castillo, L. P., and Wrobel, S. 2003. Learning minesweeper with multirelational learning. In *IJCAI*, 533–540.

Collet, R. 2005. Playing the minesweeper with constraints. In Van Roy, P., ed., *Multiparadigm Programming in Mozart/Oz*, volume 3389. Springer Berlin Heidelberg. 251–262.

Golan, S. 2011. Minesweeper on graphs. *Applied Mathematics and Computation* 217(14):6616–6623.

Golan, S. 2014. Minesweeper strategy for one mine. *Applied Mathematics and Computation* 232:292–302.

Kamenetsky, D., and Teo, C. H. 2007. Solving minesweeper using graphical models. Technical report, Australian National University.

Kaye, R. 2000. Minesweeper is np-complete. *The Mathematical Intelligencer* 22(2):9–15.

Kocsis, L., and Szepesvári, C. 2006. Bandit based montecarlo planning. In *European conference on machine learning*, 282–293. Springer.

Legendre, M.; Hollard, K.; Buffet, O.; and Dutech, A. 2012. Minesweeper: Where to probe? Technical report, INRIA.

Pedersen, K. 2004. The complexity of minesweeper and strategies for game playing. Technical report, Warwick University.

Pena, M. L. 2004. *Search Improvements in Multirelational Learning*. Ph.D. Dissertation, PhD thesis, Ottovon-Guericke-Universität Magdeburg.

Ramsdell, J. D. 1995. Programmer's minesweeper. http://www.ccs.neu.edu/home/ramsdell/pgms/.

Sebag, M., and Teytaud, O. 2012a. Upper confidence tree-based consistent reactive planning application to minesweeper. In *Learning and Intelligent Optimization*. Springer. 220–234.

Sebag, M., and Teytaud, O. 2012b. Combining myopic optimization and tree search: Application to minesweeper. In *LION6, Learning and Intelligent Optimization*, volume 7219, 222–236. Paris, France: Sringer Verlag.

Studholme, C. 2000. Minesweeper as a constraint satisfaction problem. *Unpublished project report*.