2012 Conference on Technologies and Applications of Artificial Intelligence

# Solving the Slitherlink Poblem

Tung-Ying Liu[1], I-Chen Wu[1], Der-Johng Sun[1]
[1] Department of Computer Science, National Chiao Tung University
Hsinchu, Taiwan
{tyliu,icwu,derjohng}@java.csie.nctu.edu.tw

***Abstract* — Slitherlink is one of challenging puzzle games to human and computer players. In this paper, we propose an efficient method to solve Slitherlink puzzles. After using this method, we can solve each of 10,000 25x30 puzzles given in [9] within 0.05 seconds. Without using the method, it takes at least 10 minutes to solve some of these puzzles.**

***Keywords: Slitherlink, NP-completeness, puzzle games***

## I. Introduction

*Slitherlink*, first appearing in Puzzle Communication Nikoli [6] in 1989, is one of most popular puzzle games, listed in [9]. A *Slitherlink puzzle* is played on a given rectangular grid with $m \times n$ squares and given hints on some squares. Initially, for all squares, only vertices and hints are shown, while border edges of squares (simply called *edges* in this paper) are not. For example, a $5 \times 5$ Slitherlink puzzle is shown in Figure 1 (a).

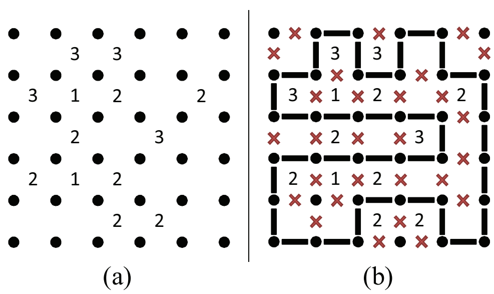

Figure 1: (a) A Slitherlink puzzle and (b) its solution.

The goal of solving a Slitherlink puzzle is to draw some of these edges such that these drawn edges form one and only one simple cycle with no intersections and with the following constraint. For each hint of a square, the hint value is the same as the number of drawn edges around the square. For example, a solution of the puzzle in Figure 1 (a) is given in Figure 1 (b). In all figures of this paper, red "×"s on edges, called *cross-out edges* or *XT edges*, indicate not to draw, while solid lines indicate to draw and blank are undecided yet.

Solving a Slitherlink puzzle efficiently is challenging, especially for those with large sizes. The general problem of determining whether a Slitherlink puzzle has a solution is NP-complete like many other puzzle problems, proved by Yato [11]. He also proved that *Another Solution Problem* (*ASP*) of Slitherlink is also NP-complete [12]. ASP of a NP-complete problem is to determine whether there exists another solution (the second) after the first is found.

When compared with other puzzles such as Nonograms [4][10], Sudoku [5], Slitherlink is more challenging in the following sense. For example, for a Nonogram puzzle with $m \times n$ squares, the space complexity is $2^{mn}$. However, for a Slitherlink puzzle with $m \times n$ squares, the space complexity is $2^{2mn+m+n}$.

Some basic and simple rules of solving Slitherlink are given in [8], and also quickly reviewed in Section II. Furthermore, Herting explored most cases in $2 \times 2$ squares and put them into a table to solve puzzles quickly.

In this paper, we propose an efficient method to identify the number of edges at a corner. Using the method, we can efficiently solve many large puzzles, such as all the $25 \times 30$ puzzles, published in [9], within 0.05 seconds.

In the rest of this paper, Section II describes some traditional methods and Herting's method for solving Slitherlink puzzles, and Section III proposes our new methods. Experiments are done in Section IV and concluding remarks are given in Section V.

## II. Traditional Methods

In this section, we first review some basic methods in Subsection II.A, and then describe Herting's method in Subsection II.B.

### *A. Basic Methods*

Starting from an empty grid, we draw and cross out edges following some deduction rules. A grid is called *a partially solved grid*, until all edges are drawn or crossed out. In this Subsection, we review some basic deduction rules in [8] to solve Slitherlink puzzles. Following are two basic deduction rules related to the maintenance of one simple cycle.

First, for each vertex, the number of drawn edges incident to it is either 0 or 2. Since a solution must contain one and only simple cycle, the cycle either goes through the vertex or does not. The number of drawn edges incident to it is 2 for the former, and 0 for the latter.

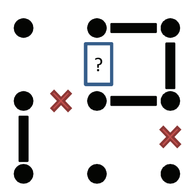


Figure 2: An example of partially solved puzzle.

978-0-7695-4919-4/12 $26.00 © 2012 IEEE
DOI 10.1109/TAAI.2012.36

Second, the edge is XT, if drawing it forms one simple cycle but there still exists some other drawn edges. For example, for a partially solved grid in Figure 2, the edge marked with blue rectangle must be XT; otherwise, a simple cycle is formed and there is still another drawn edge in the lower left.

The rest of deduction rules in this subsection are related to the hint values. First, it is trivial and omitted for the cases for the hint zero and four[1].

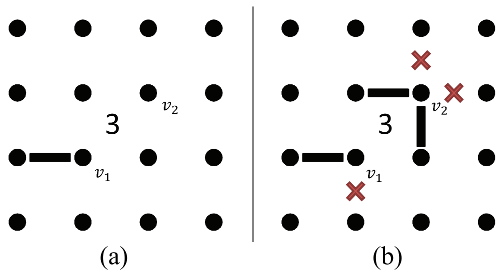


Figure 3: (a) A partially solved grid and (b) its deduced grid.

Second, consider the squares with hint three. One of the deduction rules in [8] is illustrated by the case in Figure 3 (a), where the left edge[2] of $v_1$ is drawn. If the down edge of $v_1$ is drawn, then both edges upward and rightward must be XT from the above deduction rules, contradictory to the hint three. Therefore, the down edge of $v_1$ must be XT and either up and right edge is drawn. This implies that the up and right edges of the square with hint three must be drawn. Again, the rest of two edges incident to $v_2$ are XT. The deduced grid is shown in Figure 3 (b).

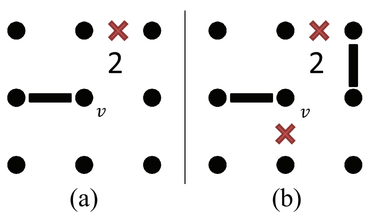


Figure 4: (a) A partially solved grid and (b) its deduced grid.

Third, consider the squares with hint two. One of the deduction rules in [8] is illustrated by the case in Figure 4 (a), where the left edge of $v$ is drawn and the up edge of the square with hint two is XT. Using a similar method, we can deduce that the down edge of $v$ is XT and the right edge of the square is drawn.

Finally, consider the squares with hint one. One of the deduction rules in [8] is illustrated by the case in Figure 5 (a) (below), where the left edge of $v$ is drawn and the down edge is XT. Using a similar method, we can deduce that both up and right edges of the square are XT.

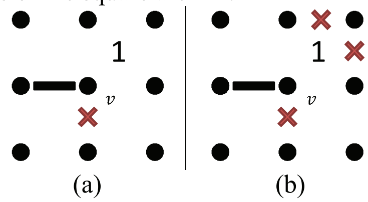


Figure 5: (a) A partially solved grid and (b) its deduced grid.

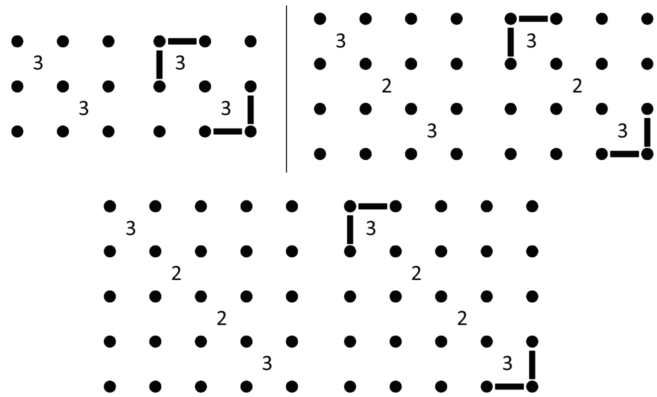


Figure 6: Partially solved grids and their deduced grids.

Several deduction rules with multiple hints were introduced in [8]. Most of these rules work for squares in the same diagonal (of different sizes) as illustrated in Figure 6, where two hint-three squares are in the same diagonal, and all the other squares are hint-two and between the two squares. In [8], they also gave some more deduction rules with different hints in the same diagonal.

### B. *Herting's Method*

In [2], Herting developed two rule-based methods to solve Slitherlink puzzles for his program. First, he used three sets of rules to help:

- All the *useful* rules for all $1 \times 1$ squares. The combination of all $1 \times 1$ squares is $3^4 \times 5 = 405$, where 3 indicates three kinds of edges, drawn, XT, and unknown, 4 indicates the four edges around the square, and 5 indicates five kinds of hints, 0, 1, 2, 3, and no hint. However, only 115 of them are useful in the sense that some deductions can be applied.
- All the 15632 useful rules for all $2 \times 2$ squares with four different square hints and four edges in the middle. The combination is $3^4 \times 5^4 = 50625$.
- All the 47601 useful rules for all $2 \times 2$ squares with 12 different edges. The combination is $3^{12} = 531441$.

He refined the three sets to get about 200 rules out of over 60000 by removing some redundancies.

Second, he also used a method, called *Trial and generalization*. The method is to follow the following steps:

1. Find an unknown edge from a partially solved grid.

[1] Note that for the hint value four it will form a simple cycle itself. According to Slitherlink, there is one and only one simple cycle, so no other edges are allowed to be drawn. Such a puzzle is not interesting, and therefore a hint four is usually not allowed.

[2] The left edge of $v_1$ is the edge incident to $v_1$ from left. Similarly, for the right, up and down edges of $v_1$.

2. Draw this edge and run the result with the (three) rule sets. If a conflict is found, the edge must be XT.
3. Cross out this edge and run the result with the rule sets. If a conflict is found, the edge must be drawn.
4. If no conflicts are found from the above two, compare the two results to draw and cross out some edges.
5. If the partially solved grid has been updated, repeat the above again.

## III. Our New Method

In addition to the traditional methods, we proposed two new methods to solve Slitherlink puzzle efficiently. The first method is to build a table of all $2 \times 2$ squares exhaustively (and part of $3 \times 3$ squares). The second method is to check corners efficiently. Due to the limit of paper size, the first method is ignored in this paper. This section focuses on the second method. We first give our notation in Subsection III.A. All the deduction rules used in the method is described in Subsection III.B. Finally, we show some cases that the traditional methods cannot solve but the method can in Subsection III.C.

### *A. Definitions and Notation*

In our method, we check each corner of two adjacent edges. For a square $S$, let $C_{DR}(S)$, $C_{DL}(S)$, $C_{UR}(S)$, $C_{UL}(S)$ be the down-right, down-left, up-right, up-left corners of $S$. For a corner $C$, say $C_{DR}(S)$, $C_{UL}(S)$ is its *diagonal corner* and both $C_{DL}(S)$ and $C_{UR}(S)$ are its *adjacent corners* with respect to $S$. These corners are illustrated in Figure 7 (a).

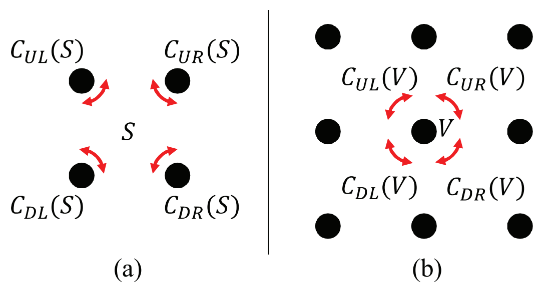


Figure 7: (a) Corners of $S$ (b) and corners of $V$.

For a vertex $V$, let $C_{DR}(V)$, $C_{DL}(V)$, $C_{UR}(V)$, $C_{UL}(V)$ be the down-right, down-left, up-right, up-left corners of $V$. For a corner $C$, say $C_{DR}(V)$, $C_{UL}(V)$ is its diagonal corner and both $C_{DL}(V)$ and $C_{UR}(V)$ are its adjacent corners with respect to $V$. These corners are shown in Figure 7 (b).

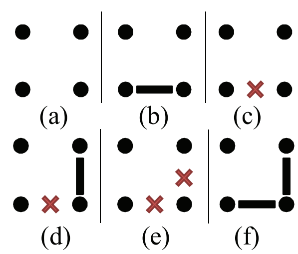


Figure 8: Six $1 \times 1$ squares.

For a corner $C$, the edge count set $E(C)$ is the set of possible numbers of drawn edges at $C$. Since the number of drawn edges at a corner is at most two, the set $E(C)$ must be a subset of {0, 1, 2}. As illustrated in Figures 8 (a) to (f), the set $E(C_{DR}(S))$ for square $S$ is {0, 1, 2}, {1, 2}, {0, 1}, {1}, {0}, and {2}, respectively.

### *B. Deduction Rules*

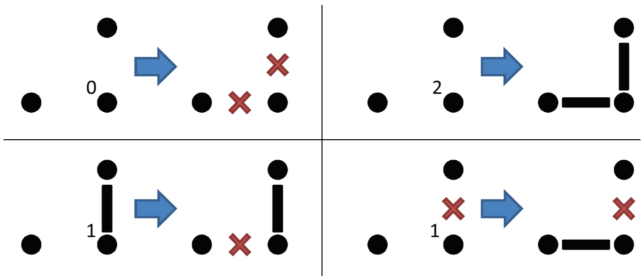


Figure 9: Some simple deduction rules.

In this subsection, we discuss all deduction rules based on corners. A set of very simple deduction rules from $E(C)$ are shown in Figure 9.

Furthermore, we investigate the deduction rules updating $E(C)$ from square hints or from one corner to others. Five situations are considered as follows.

First, assume that $S$ is a square with no hint. Then, all the four edge count sets, $E(C_{DR}(S))$, $E(C_{DL}(S))$, $E(C_{UR}(S))$, and $E(C_{UL}(S))$, are {0, 1, 2} initially.

Second, assume that square $S$ is a square with hint 0. Then, $E(C_{DR}(S))$, $E(C_{DL}(S))$, $E(C_{UR}(S))$, and $E(C_{UL}(S))$ all are {0} initially.

Third, assume that square $S$ is a square with hint 1. Then, $E(C_{DR}(S))$, $E(C_{DL}(S))$, $E(C_{UR}(S))$, and $E(C_{UL}(S))$ all are {0, 1} initially because we cannot draw more than 2 edges around a square with hint 1.

| $E(C)$ | $E(C^D)$ | $E(C^A)$ |
|---|---|---|
| {0} | Remove {0} | No change |
| {1} | Remove {1} | No change |

Table 1: Deduction rules for squares $S$ with hint 1. $C^D$ denotes a diagonal corner of $C$ and $C^A$ denotes an adjacent corner in $S$.

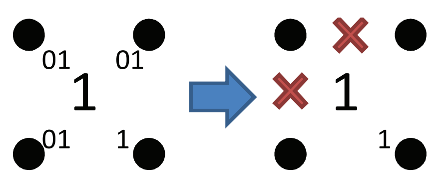


Figure 10: Example of deduction rules of square with hint 1.

In addition to the initial settings, more deduction rules for square with hint 1 are shown in Table 1. In this table, if a corner $C$ has $E(C) = \{0\}$, we can remove {0} from $E(C^D)$ and does not change $E(C^A)$, and similarly for {1}. As illustrated in Figure 10, if $E(C_{DR}(S))$ is changed to {1}, remove {1} from its diagonal corner set, $E(C_{UL}(S))$, and therefore set both upper and left edges to XT.

| $E(C)$ | $E(C^D)$ | $E(C^A)$ |
|---|---|---|
| {0} | Remove {0, 1} | Remove {0, 2} |
| {1} | Remove {0, 2} | No change |
| {2} | Remove {1, 2} | Remove {0, 2} |

Table 2: Rules of square with hint 2

Fourth, assume that square $S$ is a square with hint 2. Then, $E(C_{DR}(S))$, $E(C_{DL}(S))$, $E(C_{UR}(S))$, and $E(C_{UL}(S))$ all are {0, 1, 2} initially. In addition to the initial settings, more deduction rules for square with hint 2 are shown in Table 2.

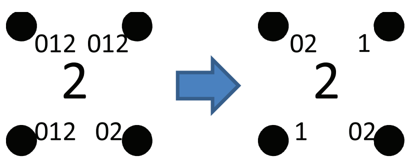


Figure 11: Example of deduction rules of square with hint 2.

An example is illustrated in Figure 11, where $E(C_{DR}(S))$ is set to {0, 2}. Assume that the number of edges at $C_{DR}(S)$ is 0. From the rule in the first row of the table (for {0}), we remove {0, 1} from its diagonal set $E(C_{UL}(S))$, and {0, 2} from its adjacent set $E(C_{DL}(S))$ and $E(C_{UR}(S))$. Then, $E(C_{UL}(S))$ is {2} and both $E(C_{DL}(S))$ and $(C_{UR}(S))$ are {1}. Now, assume that the number of edges at $C_{DR}(S)$ is 2. From the rule in the table, we obtain $E(C_{UL}(S))$ is {0} and both $E(C_{DL}(S))$ and $E(C_{UR}(S))$ are {1}. Hence, after merging the two cases by union, we obtain $E(C_{UL}(S))$ is {0, 2} and both $E(C_{DL}(S))$ and $E(C_{UR}(S))$ are still {1}. In general, in the case that $E(C)$ has multiple elements, $E(C^D)$ is the union of the results for each element, and similarly for $E(C^A)$.

| $E(C)$ | $E(C^D)$ | $E(C^A)$ |
|---|---|---|
| {1} | Remove {1} | No change |
| {2} | Remove {2} | No change |

Table 3: Rules of square with hint 3

Fifth, assume that square $S$ is a square with hint 3. Then, $E(C_{DR}(S))$, $E(C_{DL}(S))$, $E(C_{UR}(S))$, and $E(C_{UL}(S))$ all are {1, 2} initially. More deduction rules for square with hint 1 are shown in Table 3, similar to Table 1.

| $E(C)$ | $E(C^D)$ | $E(C^A)$ |
|---|---|---|
| {0} | Remove {1} | Remove {2} |
| {1} | Remove {0, 2} | No change |
| {2} | Remove {1, 2} | Remove {0, 2} |

Table 4: Deduction rules of corners $C$ at all vertices $V$. $C^D$ denotes a diagonal corner of $C$ and $C^A$ denotes an adjacent corner for $V$.

Finally, we investigate the deduction rules for corners of all vertices shown in Table 4. If $E(C)$ is {2}, $E(C^D)$ must be {0}; if $E(C)$ is {1}, $E(C^D)$ must be {1}; and if $E(C)$ is {0}, $E(C^D)$ is {0, 2}, since it is possible to have no drawn edges through the vertex. Thus, $E(C^A)$ can be deduced accordingly.

### C. *Illustration*

In this section, we illustrate 6 cases for our new method. Among them, the last two cases cannot be solved by the traditional methods in [8], described in Section II.A. For simplicity of discussion, let $V_{DR}(S)$, $V_{DL}(S)$, $V_{UR}(S)$, $V_{UL}(S)$ denote the down-right, down-left, up-right, up-left vertices of $S$, and $S_{DR}(V)$, $S_{DL}(V)$, $S_{UR}(V)$, $S_{UL}(V)$ be the down-right, down-left, up-right, up-left squares of $V$.

The first four cases are those shown in Figures 3, 4, 5 and 6. The fifth case is simple, but not listed in [8]. The sixth case demonstrates that our method can also support deduction rules on the squares not in the same diagonal. The last two demonstrates our method is general and effective.

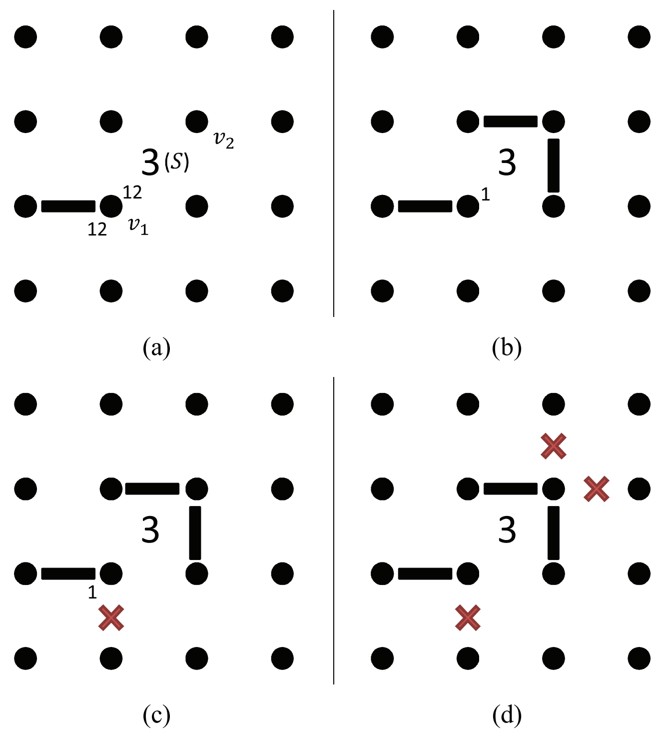


Figure 12: The first case for illustration.

For the first case shown in Figure 3, Figure 12 shows how our method can easily deduce the result in Figure 3(b). Let $S$ denote the square with hint 3. Let $v_1$ denotes $V_{DL}(S)$, and $v_2$ denotes $V_{UR}(S)$. For this rule, we know that for all corners $c$ of $S$, $E(c)$ are {1, 2} and $E(C_{UR}(v_1))$ and $E(C_{DL}(v_1))$ are {1, 2} initially in Figure 12(a), with Table 4. Then, $E(C_{DL}(S))$ will be changed to {1} by removing {2} and $E(C_{UR}(S))$ will become {2} by Table 3. So we draw the up and right edge of $s_3$ in Figure 12(b). With Table 4, $E(C_{DL}(v_1))$ becomes {1} in Figure 12(c) and $E(C_{UR}(v_2))$ become {0}. After some deductions, we obtain the one in Figure 12(d).

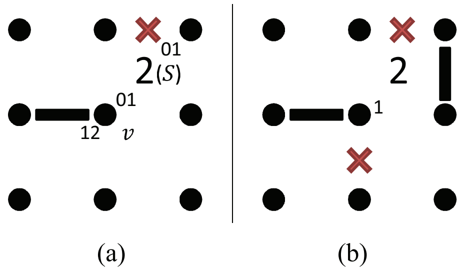


Figure 13: The second case for illustration.

For the second case shown in Figure 4, Figure 13 shows how our method can easily deduce the result in Figure 4(b). Let $S$ denote the square with hint 2. Let $v$ denotes $V_{DL}(S)$. We obtain that $E(C_{UR}(S))$ is {0, 1} and $E(C_{DL}(v))$ are {1, 2} initially. With $v$ and Table 4, we change $E(C_{DL}(S))$ to become {0, 1} in Figure 13(a). With Table 2, change $E(C_{UR}(S))$ to {1} due to $E\big(C_{DL}(S)\big)$ and then go back to change $E\big(C_{DL}(S)\big)$ to {1}. Then, from Table 4, $E\big(C_{DL}(v)\big)$ become {1}. So, we draw the right edge of $S$ and cross out the down edge of $v$ as in Figure 13(b).

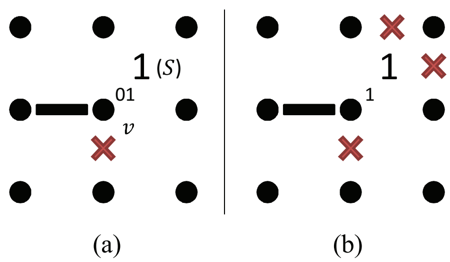


Figure 14: The third case for illustration.

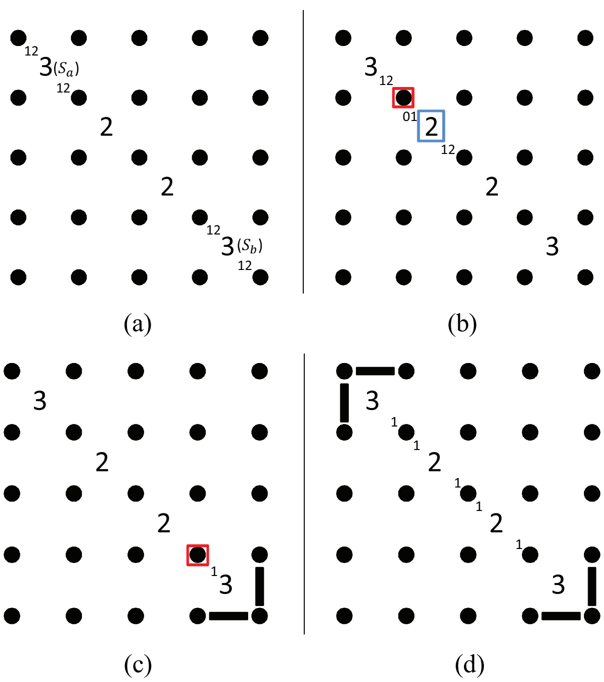


Figure 15: The fourth case for illustration.

For the third case shown in Figure 5, Figure 14 shows that our method can easily deduce the result in Figure 5(b). This is similar to the above case, and therefore omitted. Similarly, for the fourth case shown in Figure 6, Figure 15 shows that our method can easily deduce the results in Figure 6.

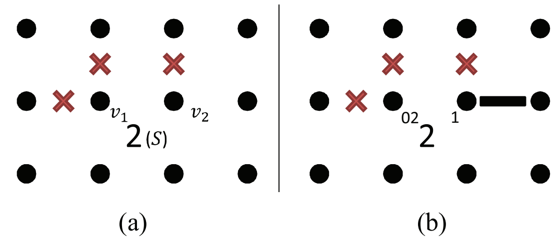


Figure 16: The fifth case for illustration.

The fifth case is shown in Figure 16. Let $S$ denote the square with hint 2. Let $v_1$ denote $V_{UL}(S)$ and $v_2$ denote $V_{UR}(S)$. With Table 4, $E\big(C_{UL}(S)\big)$ is changed to {0, 2} based on $v_1$. Then, from Table 2, $E\big(C_{UR}(S)\big)$ is changed to {1} based on $E\big(C_{UL}(S)\big)$; from Table 4, $E\big(C_{UR}(v_2)\big)$ is set to {1} based on $E\big(C_{UR}(S)\big)$. So the right edge of $v_2$ is drawn as in Figure 16 (b).

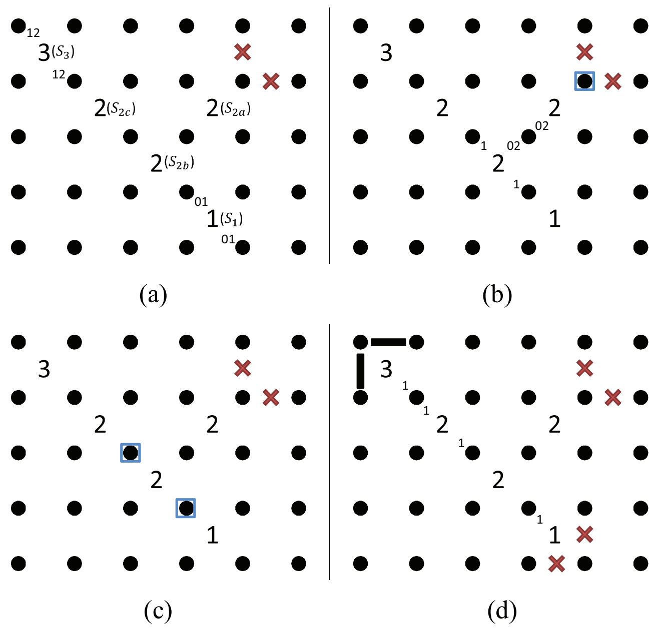


Figure 17: The sixth case for illustration.

Finally, the sixth case is shown in Figure 17, which can demonstrate how general and effective our method is. Let $S_3$ denote the square with hint 3 and $S_1$ denote the square with hint 1. Let $S_{2a}$, $S_{2b}$ and $S_{2c}$ denote three squares with hint 2. Initially, for all corners $c_3$ of $S_3$, $E(c_3)$ are {1, 2}, and for all corners $c_1$ of $S_1$, $E(c_1)$ are {0, 1} in Figure 17(a). Let $v_1$ denote the vertex in blue rectangle in Figure 17(b). From Table 4, since $E\big(C_{UR}(v_1)\big)$ is {0}, $E\big(C_{DL}(v_1)\big)$ becomes {0, 2}. From Table 2, $E\big(C_{DL}(S_{2a})\big)$ also becomes {0, 2}. From

Table 2, since $E(C_{UR}(S_{2b}))$ is {0, 2}, both $E(C_{UL}(S_{2b}))$ and $E(C_{DR}(S_{2b}))$ are {1} as in Figure 17(b). Then, the two sets are propagated to $S_1$ and $S_3$, respectively, and subsequently cross out two edges of $S_1$ and draw two edges of $S_3$, respectively, as in Figure 17(d).

## IV. EXPERIMENTS

In our experiments, we used the personal computer equipped with the CPU, Intel(R) Core(TM)2 Duo E7200 @ 2.53GHz, to solve 10000 25×30 puzzles ranked hard in [9]. We compare the program, denoted by $P_{new}$, using the method in Subsection III with another program, denoted by $P_{old}$. The program $P_{old}$ uses the deduction rules in [8], not including those in Subsection III.

In the program $P_{old}$, at least 2% of the 10000 puzzles cannot be solved within 2000 seconds. After using $P_{new}$, all can be solved within 0.05 second.

We also compare our program with Herting's in solving $40 \times 35$ puzzle given in [1]. For Herting's method, it took about 2 and half minutes to solve it. Our program $P_{new}$ can solve the puzzle in 1.7 seconds.

## V. CONCLUSION

In this paper, we have proposed a new method to solve Slitherlink puzzles. Unlike traditional methods and Herting's rule-based method, our new method use sets of corners to solve Slitherlink puzzle efficiently. With square hints, the new method supports many efficient and simple deduction rules. In our experiments, the new method performs extremely better than the old version. In addition, we can also solve a $40 \times 35$ puzzle in [1], which requires 2 and half minutes using Herting's method.

In addition, since our method can solve all the current Slitherlink puzzles very quickly, we are building a puzzle generator to create much harder problems, and will publish over the Internet soon. The concepts in this paper may be applied to solving some other games, e.g., those in [3][7].

## REFERENCES


[1] Hirofumi Fujiwara, collection of slitherlink puzzles, http://www.pro.or.jp/~fuji/java/puzzle/numline/index-eng.html.

[2] Stefan Herting, A rule-based approach to the puzzle of Slither Link, 2005. Available at https://www.cs.kent.ac.uk/pubs/ug/2005/co620/slither/report.pdf.

[3] Kuo-Yuan Kao, I-Chen Wu, and Yi-Chang Shan, XT Domineering: A New Combinatorial Game, Knowledge-Based Systems, Vol. 34, pp. 55-63, October 2012.

[4] H. H. Lin, D.J. Sun, I.C. Wu, and S.J. Yen, "The 2011 TAAI Computer-Game Tournaments," *ICGA Journal*, Vol. 34, No.1, pp.51-54, 2011.

[5] Hung-Hsuan Lin, I-Chen Wu, "An Efficient Approach to Solving the Minimum Sudoku Problem", ICGA Journal (SCI), vol. 34(4), pp. 191-208, December 2011.

[6] Nikoli Puzzles – slitherlink, www.nikoli.co.jp/en/puzzles/slitherlink.html.

[7] Yi-Chang Shan, I-Chen Wu, Hung-Hsuan Lin, and Kuo-Yuan Kao, "Solving Nine Layer Triangular Nim", Journal of Information Science and Engineering (SCI), vol.28, No.1, pp.99-113, January, 2012.

[8] Slitherlink, Wikipedia, http://en.wikipedia.org/wiki/slitherlink.

[9] Slither Link – online puzzle game, http://www.puzzle-loop.com.

[10] I-Chen Wu, Der-Johng Sun, Kan-Yueh Chen, Ching-Hua Kuo, Hao-Hua Kang and Hung-Hsuan Lin, A Highly Efficient Nonogram Solver, in preparation, 2012.

[11] Takayuki Yato, On the NP-completeness of Slither Link Puzzle, IPSJ SIGNotes ALgorithms, 2000.

[12] Takayuki Yato, Complexity and completeness of finding another solution and its application to puzzles, IEICE TRANSACTIONS on Fundemantals of Electronics, Communications and Computer Sciences Vol.E86-A No.5 pp.1052-1060, 2003