2005－GI－14（5）
2005／9／5

Automatic Evaluation on Difficulty of Logic-Puzzle Using Heuristics

Takeshi Ito
Department of Computer Science, The University of Electro-Communications
ito@cs.uec.ac.jp

In the one-player game like a puzzle, generally the generators and the solvers of the puzzle exist and a kind of community is formed. When many puzzle problems can be generated with a fixed rule, these problems are called “works”, and there are good or bad and a degree of difficulty in these problems. A work tends to be evaluated by a degree of difficulty and artistic effect in such a community, and especially difficulty is useful information for solvers.

In this research, I investigated how the expert would solve the problem on a Logic Puzzle. As a result, the expert transformed the logical solutions into the heuristics solutions by the ease of applying and solved the problems by using the heuristics was observed. Furthermore, when investigated in detail, it turned out that it is classified into the 23 rules and 6 heuristics.

All of the 23 rules and 6 heuristics were implemented in the computer. Furthermore, I investigated the order which the expert applies and introduced the automatic solving system applied in order of the ease of applying. And I applied the same problem answer to experts and the computer for comparing answer time of experts and use frequency of 6 heuristic. After that, I identified the elements which are related to a degree of difficulty and suggested a system evaluating a degree of difficulty automatically.

ヒューリスティックスを用いたロジックパズルの難易度自動評価

伊藤　毅志
電気通信大学情報工学科
ito@cs.uec.ac.jp

パズルのような一人ゲームでは、一般にパズルの作者と回答者が存在し、一種のコミュニティーを形成している。一定のルールで複数のパズル問題が生成できる場合、作られた問題は作品と呼ばれ、良し悪しや難易度が生じる。こういったコミュニティーでは、作品は難易度や芸術性から評価される傾向があり、特に難易度は回答者にとって有用な情報であることが多い。本研究では、ロジックパズルを題材にして、パズルの熟達者がどのように問題を解いているのかを調査した。その結果、これら熟達者は、論理的な解法を適用しやすさから経験的な解法に変形して、解決していることが観察された。さらに詳しく調べると、２３のルールと６つのヒューリスティックスに分類されることがわかった。

ここでは、これら２３のルールと６つのヒューリスティックスをすべてコンピュータに組み込んだ。さらに、熟達者が適用する順を調査して、適用しやすさの順に適用する自動回答システムを構

築した。そして、熟達者とコンピュータに同じ問題を回答させて、熟達者の回答時間と６つのヒューリスティックスの利用頻度を比較して、難易度に関係しているヒューリスティックスを同定して、難易度を自動評価するシステムを提案する。

1. Introduction

In the one-player game like a puzzle, generally the generators and the solvers of the puzzle exist and a kind of community is formed. When many puzzle problems can be generated with a fixed rule, these problems are called "works", and there are good or bad and a degree of difficulty in these works. A work tends to be evaluated by a degree of difficulty and art characteristics in such a community, and especially difficulty is useful information for solvers. From this, it is necessary to evaluate a work. However, the skillful person of the puzzle actually solved the problem, and many puzzles have given difficulty and art as evaluation experientially. In order to evaluate, it is necessary to actually solve a problem with a help, and a great labor is needed for this. In addition, there is the problem that there is the evaluation unevenness so that subjectivity of a solver gets.

In order to solve such a problem, research which develops the system by which the art and the difficulty of a work are automatically evaluated for an objective numerical value has so far been done. By research dealing with a Tume-Shogi problem, there are some which proposed the standard which evaluates the art of the problem by giving a score to a move [1]. The valuation basis of a Tume-Shogi problem is very various, and freshness, unexpected nature, difficulty, etc. are seen from various angles. In this research, based on the technique [2] which Murayama has proposed, a score is given to possible moves exist and the number of pieces currently used and the number of immobility pieces, the form of positions, etc., respectively, and the value which divided by moves what added them all is made into the valuation basis.

In another research dealing with Number-Place, the technique of evaluating the difficulty in problem is proposed [3]. Number-Place is a puzzle which buries the number to 1-9 so that one number may not come out twice to each sequence, each line, and each block (squares of 3x3). If this problem is solved by computer, it can be solved by all search In this research, the point that man solved using the heuristics of limiting the number which should go into each square was noted. Counting how many times the heuristics was used in the lengthwise direction, the transverse direction, and the block, investigating the number of times of use of the solution by the combination of a block, a lengthwise direction, and a transverse direction, or comparing execution time depending on how undoing a computer, in order to solve the problem is estimating difficulty numerically. I aimed at the development of the system which evaluates difficulty of Logic Puzzle by using the technique in Number-Place automatically in this report.

2. Logic Puzzle

2.1 Rule of Logic Puzzle

Logic Puzzle will be the puzzle called “Illustration Logic”, ”Painting Logic”, “Nono-gram”, etc., and if only the part of the number of length and width smears away the squares continuously, a picture will appear.(Figure 1)

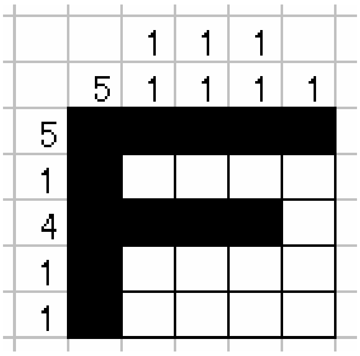

Figure 1 An example of Logic-Puzzle (standard monochrome problem)

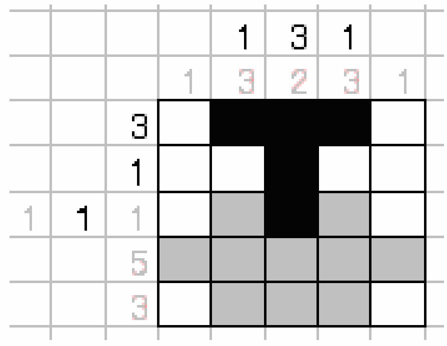

Figure 2 An example of Color-Logic (extended color problem)

A logic puzzle is divided into two kinds, monochrome logic which applies black according to only the number of numbers as shown in Figure 1, and the color logic which applies two or more colors according to the number of two or more colors as shown in Figure 2.

The rule of monochrome logic is as follows.

(1) Paint over only length of a vertical or a horizontal number consecutively

(2) In the case of the adjacent number, one or more squares vacate and apply

As a rule of color logic, it is as follows.

(1) Paint over only length of a vertical or a horizontal number consecutively by the appointed color

(2) When the number which each sequence adjoined is the same color, one or more squares vacate and apply

(3) When it is the color from which the number which each sequence adjoined is different, zero or more squares vacate and apply

Although color logic has the advantage that the completed picture is beautiful for appearance, since the place into which it goes for every color is limited, it is known that the color logic is generally easier than monochrome logic. In this report, I researched only monochrome logic which can be defined by simple rules.

2.2 A Method of Logical Solving

It turns out that a logical solution is divided into three steps of the following levels by whether a grid color is decided definitely.

Level 1: This level is a solution which can determine which shall be definitely applied to a square between white or black according to the rule of a logic puzzle.

Level 2: This level is a solution which investigates whether inconsistency arises by assuming one of the squares which cannot be determined definitely to be white or black.

Level 3: This level is a solution which investigates whether inconsistency arises by assuming simultaneously two or more squares which cannot be determined definitely to be white or black.

There are the following as a typical example of a solution on level 1. It is the method of assuming what was brought near by the rightmost and by the leftmost based on the number of each line and determining the portion of the intersection as black or white. All the squares are definitely determined as black or white by performing this to a line and a sequence repeatedly.

For example on the problem of Figure 1, the 3rd line from the top is determined by an intersection of brought near by rightmost and by leftmost (see Figure 3).

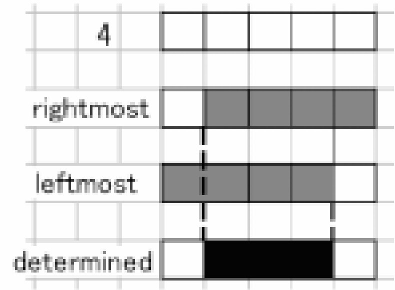

Figure 3 An example of typical solution of logic puzzle

All the logical solutions of the level 1 are constituted by the solution derived from this typical solution.

3. Heuristics

3.1 Observation of expert

I selected two expert players in a logic puzzle. The thought processes of the subjects were examined by the data from video recorded protocols when solving the puzzles. As a result, a common solution order was observed by the process of solution of the expert players. It is considered to be based on the solution heuristics which the expert players have in common. It has turned out that the heuristics is formed in the form of rules for an expert player to solve and the priority of the ease of applying the rules. Furthermore, as a result of investigating in detail, the rule which the expert players have was 23 in all, and when classified from the ease of carrying out of application, being classified into 6 categories also became clear. The 6 categories are as follows.

① The rule which applies a typical solution to the number (MAX) paying attention to a bigger number than that in all the numbers (see Figure 3)

② The rule which all the squares of a line decide (see Figure 4)

- If (Length of the line) = (Σ (each n)+n-1) then this rule are applied

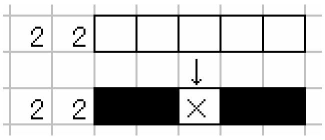

Figure 4　An example of category (2)

③ The rule which pays its attention to both ends (see Figure 5)

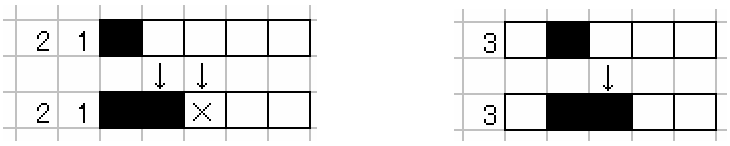

Figure 5　Two examples of category (3)

④ The rule applied when the undecided number of a line is 0 or 1 (see Figure 6)

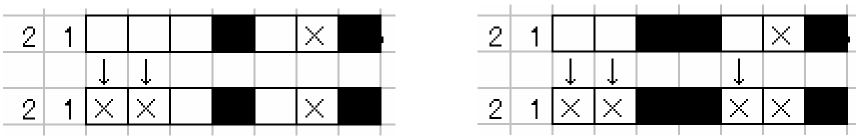

Figure 6　Two examples of category (4)

(Leftside: undecided number = 1, Rightside: undecided number = 0)

⑤ The rule which observes the maximum number or the minimum number of a line (see Figure 7)

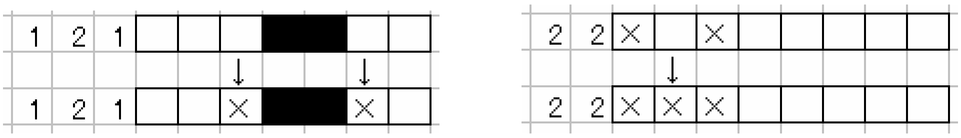

Figure 7　Two examples of category (5)

(Leftside: maximum number =2 , Rightside: minimum number =2)

⑥ The rules are not applied to the above rules (see Figure 8)

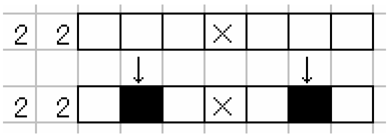

Figure 8　An example of category (6)

I applied these rules in order of this category, and implemented on the computer the system which solves a logic puzzle. Since these rules contain all the deterministic rules, all the problems solved on a level 1 are solvable. Moreover, since this system is equipped also with the function to assume a square one after another and to verify it to the problem which was not solved on a level 1, it can respond to the problem of a level 2.

4. Experiment

4.1 Objective and Method

In order to decide a valuation basis for a system to determine difficulty automatically, to man and a computer, the same problem was given and the experiment which compares each result was conducted. The problems used for the experiment are 15 problems (P1-15) of the size of 15x15 and 22 problems (Q1-22) of the size of 20x20 which are introduced to the magazine [3] and book [4] of the logic puzzle. As an experiment for man, the above problems were given to three expert subjects of a logic puzzle, the average of the answer time to solve them was measured. On the other hand, the same problems were also given to the computer, it counted what kind of rule was used how many times, and the execution process was saved. As what is expected especially when it comes to the standard of evaluation of difficulty, I paid my attention to the following data.

(A) The number of times of each rule used by the time it solved
(B) The value of **MAX** lowered by the time it solved
(C) The number (**TN**) of turns investigated by the time it solved by considering investigating a vertical and a horizontal line once as one turn
(D) The number (**UN**) of squares which was not solved in a logical solution

4.2 Results and Discussion

The data of the P-problems (15x15) are shown in Table 1, and the data of the P-problems (20x20) are shown in Table 2. The problems are rearranged into the short order by the time which man solved.

	H	①	②	③	④	⑤	⑥	MAX	TN	UN
P4	205	19	11	38	17	0	0	7	126	0
P2	213	9	17	40	17	3	4	7	114	0
P10	224	4	11	50	22	2	3	7	67	0
P1	227	4	5	65	20	3	2	7	207	0
P6	231	8	9	51	15	7	4	7	115	0
P5	263	15	6	53	17	5	5	7	90	0
P3	328	28	8	47	18	8	3	7	182	0
P7	392	1	12	63	25	7	4	7	179	0
P12	396	11	7	57	20	3	2	3	573	0
P13	410	12	12	46	17	13	10	4	504	0
P8	411	29	4	48	21	10	5	7	145	0
P14	458	26	6	69	22	3	5	7	208	0
P11	609	28	5	65	20	3	3	5	532	0
P15	829	20	12	71	18	7	7	3	756	0
P9	885	22	8	47	22	17	4	4	650	0

Table 1　The data of the P-problems

	H	①	②	③	④	⑤	⑥	MAX	TN	UN
Q14	485	13	9	74	21	1	0	10	69	0
Q7	491	17	9	109	22	10	7	10	181	0
Q18	590	46	4	88	26	22	7	4	751	0
Q1	632	27	5	104	28	9	5	3	694	0
Q15	660	9	12	124	17	3	8	10	130	0
Q22	668	40	9	85	23	16	8	5	690	0
Q16	670	9	12	119	28	8	15	6	749	0
Q10	686	25	18	83	25	15	4	7	641	0
Q11	706	6	11	127	23	19	20	9	727	0
Q8	709	36	11	75	36	9	3	6	745	0
Q4	724	24	11	100	37	14	7	5	737	0
Q20	790	11	19	124	18	9	8	7	653	0
Q2	800	32	14	101	25	13	7	3	1060	0
Q3	920									248
Q17	972	40	7	110	29	17	7	5	831	0
Q6	999									296
Q5	1012	27	9	83	32	18	5	4	868	0
Q13	1072									388
Q19	1089									272
Q9	1154									175
Q21	1325									385
Q12	1533									244

Table 2　The data of the Q-problems

The time which man solved is expressed with **H** and the unit is second. ①〜⑥ are the classifications of above heuristics. When it is as difficult a problem as the problem which required time, the value of MAX has the tendency for an easier problem to become large and for a more difficult problem to become small, from Table 1. The value of TN also has the tendency for a more difficult problem to become large. Table 2 showed that the problem that the value of UN is larger than 0 was a difficult problem in many cases. Moreover, there was no tendency in particular in the frequency in use of each heuristics. From the above data, MAX, TN, and UN are considered to be elements in connection with evaluation of difficulty.

5. Automatic Evaluation System

The system which solves a problem automatically can count MAX, TN, and UN. In order to evaluate difficulty automatically, it is necessary to give the suitable dignity for the value of MAX, TN, and UN, and to normalize. It is thought that it is difficult, so that **MAX** becomes smaller than it on the basis of (a size of a puzzle/ 2). It is thought that it is difficult, so that **TN** becomes a large value simply. If **UN** is a larger number than zero, it will be the problem included assumption and it will be thought that it is quite difficult.

Supposing it expresses difficulty in five steps in the case of P-problems (15x15), difficulty (**Dp**) is presumed by the following formulas.

$$Dp = (abs\ (abs(7 - MAX) + abs(TN / 100) + \#UN \times 8) / 2) + 1$$

(if UN = 0 then #UN = 0 else #UN =1)

Similarly, supposing it expresses difficulty in five steps in the case of Q problem, difficulty (**Dq**) will be presumed by the following formulas.

$$Dq = (abs\ (abs(10 - MAX) + abs(TN / 150) + \#UN \times 12) / 3) + 1$$

(if UN = 0 then #UN = 0 else #UN =1)

If the difficulty of the problem of P1-15 and Q1-22 is presumed using these formulas, it will become as it is shown in Table 3 and Table 4. The correlation coefficient between the answer time and presumed difficulty in Table 3 is 0.772. Moreover, the correlation coefficient between the answer time in Table 4 and presumed difficulty is 0.800. From these results, a certain amount of result of difficulty presumption by this calculation was shown.

	H	Dp
P4	205	1
P2	213	1
P10	224	1
P1	227	1
P6	231	1
P5	263	1
P3	328	1
P7	392	1
P12	396	4
P13	410	4
P8	411	1
P14	458	1
P11	609	3
P15	829	5
P9	885	4

	H	Dq
Q14	485	1
Q7	491	1
Q18	590	3
Q1	632	3
Q15	660	1
Q22	668	3
Q16	670	3
Q10	686	2
Q11	706	2
Q8	709	3
Q4	724	3
Q20	790	2
Q2	800	4
Q3	920	5
Q17	972	3
Q6	999	5
Q5	1012	3
Q13	1072	5
Q19	1089	5
Q9	1154	5
Q21	1325	5
Q12	1533	5

Table 3 Answer time and presumed difficulty on P-problems (left side)

Table 4 Answer time and presumed difficulty on P-problems (right side)

6. Conclusion

In this report, I could implement heuristics which man solves to the computer, by investigating experts' solution processes. And the element (MAX, TN, UN) in connection with difficulty was able to be extracted by comparing with evaluation of man. Furthermore, the method of presuming the difficulty of problem was introduced using the formula based on these elements. It turned out that difficulty can be presumed somewhat correctly by this method.

References

1. Hitoshi Matsubara, Ken-ichi Handa and Fumio Motoyoshi : "A Computer program of automatic evaluation of TSUME-SHOGI" , IPSJ-ICS-80, (1992), 1-10
2. Nobuo Inui and Yoshiyuki Kotani : "The solution, calculation of difficulty, the creation of problem on NUMBER PLACE" , The 7th Game Programming Workshop, (2002), 163-170
3. " OEKAKI LOGIC (Painting logic) ", SEKAIBUNKA-SHA, No.103, No.109 (2004)
4. " E NI NARU LOGIC PUZZLE (The logic puzzle which become a picture) ", NITTO-SHOIN, (2004)