

人工知能Ⅱ 課題

問題の状態空間表現

5CS 10 糸山莉奈子

1. アルゴリズムの説明

以下にアルゴリズムのフローチャートを示す。

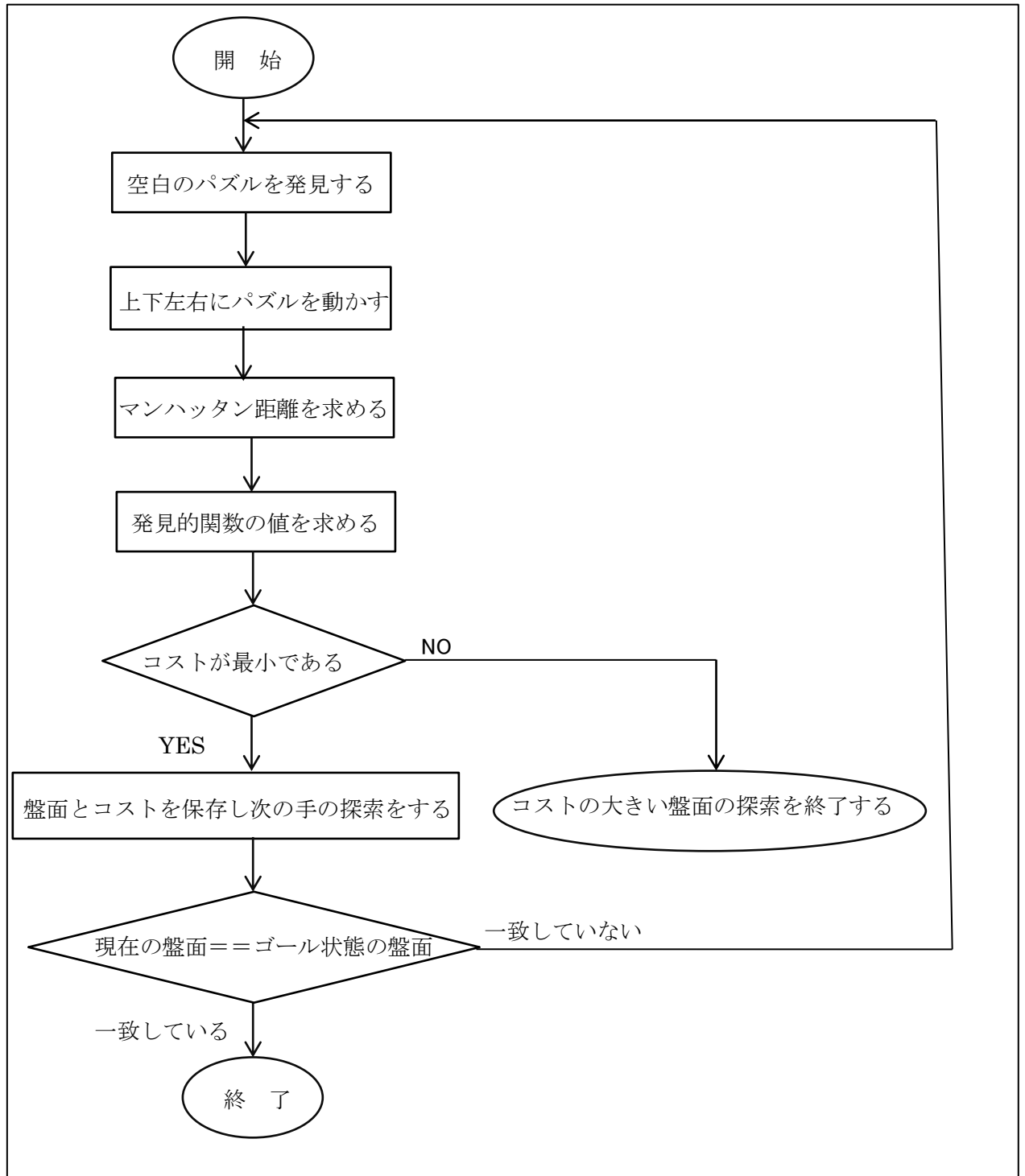


図 1. アルゴリズムのフローチャート

2. ソースコード

以下にソースコードを示す。

言語：C 言語

```
3. #include<stdio.h>
4. #include<stdlib.h>
5. #include<string.h>
6. #include<math.h>
7. #pragma warning(disable:4996)
8. #define QUEUE_SIZE 50
9.
10. //隣接リスト
11. int rin[9][6] = {
12.     1, 2, -1, -1, -1, 8, 0, 5, -1, -1, 1, 4, -1, -1, -1,
13.     8, 0, 6, -1, -1, 1, 2, 3, 4, -1, -1, 5, 0, 7, -1, -1,
14.     2, 3, -1, -1, -1, 6, 0, 7, -1, -1, 4, 3, -1, -1, -1, };
15.
16. int goal[3][3] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 0 } }; //ゴール状態
17. int cell1[3][3] = { { 8, 1, 5 },
18.                     { 2, 0, 4 },
19.                     { 6, 3, 7 } }; //初期状態
20. int q, r, j = 0;
21. //キュー用関数
22. int enqueue(int min);
23. int dequeue(int min2);
24. int queue[QUEUE_SIZE];
25. int head, tail = 0;
26. //パズルを動かすための関数
27. int moveup(int cell1[3][3]);
28. int movedown(int cell1[3][3]);
29. int moveleft(int cell1[3][3]);
30. int moveright(int cell1[3][3]);
31. //マンハッタン用関数
32. int manhattan(int cell[3][3]);
33. //最小コスト用関数
34. int mostmin(int min1, int min2, int min3, int min4);
35.
```

```

36. int hozon(int cell1[3][3]);
37. typedef int data_t;
38.
39. int moveup(int cell1[][3]) { // 0を発見して上と入れ替え
40.     int x1, y1=0;
41.     int c = 0;
42.     int con1 = 0;
43.     int c1[3][3] = { 0 };
44.
45.     for (x1 = 0; x1 <= 2; x1++) {
46.         for (y1 = 0; y1 <= 2; y1++) {
47.             if (cell1[x1][y1] == 0 && c==0) { // 上
48.                 con1 = cell1[x1 - 1][y1];
49.                 cell1[x1][y1] = con1;
50.                 cell1[x1 - 1][y1] = 0;
51.                 c++; // 一回以上入れ替わらないようにする
52.             }
53.         }
54.     }
55.     return cell1[x1][y1];
56. }
57.
58. int movedown(int cell1[][3]) { // 0を発見して下と入れ替え
59.     int x1, y1, n=0;
60.     int con2=0;
61.     int c = 0;
62.
63.     for (x1 = 0; x1 <3; x1++) {
64.         for (y1 = 0; y1 <3; y1++) {
65.             if (cell1[x1][y1] == 0 && c==0) {
66.                 con2= cell1[x1 + 1][y1];
67.                 cell1[x1 + 1][y1] = 0;
68.                 cell1[x1][y1] = con2;
69.                 c++; // 一回以上入れ替わらないようにする
70.             }
71.         }

```

```

72.     }
73.     return cell1[x1][y1];
74. }
75.
76. int moveleft(int cell1[][3]) { // 0を発見して左と入れ替え
77.     int x1, y1;
78.     int con3 = 0 ;
79.     int c = 0;
80.
81.     for (x1 = 0; x1 <= 2; x1++) {
82.         for (y1 = 0; y1 <= 2; y1++) {
83.             if (cell1[x1][y1] == 0 && c==0) { // 上
84.                 con3 = cell1[x1][y1 - 1];
85.                 cell1[x1][y1 - 1] = 0;
86.                 cell1[x1][y1] = con3;
87.                 c++; // 一回以上入れ替わらないようにする
88.             }
89.         }
90.     }
91.     return cell1[x1][y1];
92. }
93.
94. int moveright(int cell1[][3]) { // 0を発見して右と入れ替え
95.     int x1, y1;
96.     int con4 = 0 ;
97.     int c = 0;
98.
99.     for (x1 = 0; x1 <= 2; x1++) {
100.        for (y1 = 0; y1 <= 2; y1++) {
101.            if (cell1[x1][y1] == 0 && c==0) { // 上
102.                con4 = cell1[x1][y1 + 1];
103.
104.                cell1[x1][y1 + 1] = 0;
105.                cell1[x1][y1] = con4;
106.                c++; // 一回以上入れ替わらないようにする
107.            }

```

```

108.         }
109.     }
110.     return cell[x1][y1];
111. }
112.
113. int manhattan(int cell[][3]) { //マンハッタン距離を求める
114.
115.     int value[8] = { 0 };
116.     int value2 = 0;
117.     int i, j, n=0;
118.     int x1, y1 = 0;
119.     int w, t = 0;
120.
121.     for (x1 = 0; x1 <= 2; x1++) {
122.         for (y1 = 0; y1 <= 2; y1++) {
123.             n = cell[x1][y1]; //セルの中身取得
124.             //printf("n==%d\n", n);
125.             w = x1; //座標位置取得
126.             t = y1;
127.
128.             switch (n) { //マンハッタン計算(中身に応じて変動)
129.                 case 0:
130.                     break;
131.                 case 1:
132.                     q = abs(0 - w);
133.                     r = abs(0 - t);
134.                     value[0] = abs(q + r);
135.                     break;
136.
137.                 case 2:
138.                     q = abs(0 - w); //x
139.                     r = abs(1 - t); //y
140.                     value[1] = abs(q + r);
141.                     break;
142.
143.                 case 3:

```

```
144.         q = abs(0 - w); //x
145.         r = abs(2 - t); //y
146.         value[2] = abs(q + r);
147.         break;
148.
149.     case 4:
150.         q = abs(1 - w); //x
151.         r = abs(0 - t); //y
152.         value[3] = abs(q + r);
153.         break;
154.
155.     case 5:
156.         q = abs(1 - w); //x
157.         r = abs(1 - t); //y
158.         value[4] = abs(q + r);
159.         break;
160.
161.     case 6:
162.         q = abs(1 - w); //x
163.         r = abs(2 - t); //y
164.         value[5] = abs(q + r);
165.         break;
166.
167.     case 7:
168.         q = abs(2 - w); //x
169.         r = abs(0 - t); //y
170.         value[6] = abs(q + r);
171.         break;
172.
173.     case 8:
174.         q = abs(2 - w); //x
175.         r = abs(1 - t); //y
176.         value[7] = abs(q + r);
177.         break;
178.     }
179. }
```

```

180.     }
181.
182.     //マンハッタン集計-----
183.     for (i = 0; i < 8; i++) {
184.         value2 += abs(value[i]);
185.     }
186.     value2 = value2 + 1;
187.     return value2;
188.     //-----
189. }
190. int mostmin(int min1, int min2, int min3, int min4) {
191.     //マンハッタン距離の最小値の算出
192.     int min = 0;
193.
194.     min = min1;
195.     if (min>=min2) {
196.         min = min2;
197.     }
198.
199.     if (min>=min3) {
200.         min = min3;
201.     }
202.
203.     if (min>=min4) {
204.         min = min4;
205.     }
206.     printf("MIN=%d ¥n", min);
207.     return min; //最小のコストを返す
208. }
209. int enqueue(int min) { //キューに最小コスト挿入
210.     queue[tail++] = min;
211.     return 0;
212. }
213. int dequeue(int *min2) { //キューから値を取り出し
214.     *min2 = queue[head++];
215.     return 0;

```



```

216.     }
217.     int hozon(int cell1[][3]) {
218.         int c_hozon[3][3] = { 0 };
219.         for (int x1 = 0; x1 <= 2; x1++) {
220.             for (int y1 = 0; y1 <= 2; y1++) {
221.                 c_hozon[x1][y1] = cell1[x1][y1];
222.                 //printf("c_hozon = %d¥n", c_hozon[x1][y1]);
223.             }
224.         }
225.         return c_hozon;
226.     }
227.     int main(void) {
228.         int man = 0;
229.         int man2 = 0;
230.         int man3 = 0;
231.         int man4 = 0;
232.         int min = 0;
233.         int max = 0;
234.         //パズルの状態を保存
235.         int c_1[3][3] = { 0 };
236.         int c_2[3][3] = { 0 };
237.         int c_3[3][3] = { 0 };
238.         int c_4[3][3] = { 0 };
239.
240.         //while (cell1 != goal) {
241.             //上に動かす
242.             moveup(cell1);
243.             man = manhattan(cell1);
244.             //盤面保存
245.             for (int x1 = 0; x1 <= 2; x1++) {
246.                 for (int y1 = 0; y1 <= 2; y1++) {
247.                     c_1[x1][y1] = cell1[x1][y1];
248.                 }
249.             }
250.             movedown(cell1); //元に戻す

```

```

251.             printf("MAN1=%d¥n", man);
252.
253.             //下に動かす
                -----
254.             movedown(cell1);
255.             man2 = manhattan(cell1);
256.             for (int x1 = 0; x1 <= 2; x1++) { //盤面保存
257.
258.                 for (int y1 = 0; y1 <= 2; y1++) {
259.                     c_2[x1][y1] = cell1[x1][y1];
260.                 }
261.             }
262.             moveup(cell1); //元に戻す
263.             printf("MAN2=%d¥n", man2);
264.
265.             //左に動かす
                -----
266.             moveleft(cell1);
267.             man3 = manhattan(cell1);
268.             for (int x1 = 0; x1 <= 2; x1++) { //盤面保存
269.                 for (int y1 = 0; y1 <= 2; y1++) {
270.                     c_3[x1][y1] = cell1[x1][y1];
271.                 }
272.             }
273.             moveright(cell1); //元に戻す
274.             printf("MAN3=%d¥n", man3);
275.
276.             //右に動かす
                -----
277.             moveright(cell1);
278.             man4 = manhattan(cell1);
279.             for (int x1 = 0; x1 <= 2; x1++) { //盤面保存
280.                 for (int y1 = 0; y1 <= 2; y1++) {
281.                     c_4[x1][y1] = cell1[x1][y1];
282.                 }
283.             }

```

```

284.         moveleft(cell1); //元に戻す
285.         printf("MAN4=%d¥n", man4);
286.         //〜パズルの移動ここまで

```

```

287.         //最小コストを求めてminに入れる
288.         min = mostmin(man, man2, man3, man4);
289.
290.         if (min == man) { //man1が最小コストだった時
291.             //最小コストをスタックに入れる
292.             printf("最小はman1¥n");
293.             enqueue(min);
294.             hozon(c_1);
295.             for (int x1 = 0; x1 <= 2; x1++) {
296.                 for (int y1 = 0; y1 <= 2; y1++) {
297.                     cell1[x1][y1] = c_1[x1][y1];
298.                 }
299.             }
300.         }
301.         if (min == man2) { //man2のとき
302.             //最小コストをスタックに入れる
303.             enqueue(min);
304.             hozon(c_2);
305.             for (int x1 = 0; x1 <= 2; x1++) {
306.                 for (int y1 = 0; y1 <= 2; y1++) {
307.                     cell1[x1][y1] = c_2[x1][y1];
308.                 }
309.             }
310.         }

```



```

311.         if (min == man3) {
312.             //最小コストをスタックに入れる
313.             enqueue(min);
314.             hozon(c_3);
315.             for (int x1 = 0; x1 <= 2; x1++) {
316.                 for (int y1 = 0; y1 <= 2; y1++) {

```

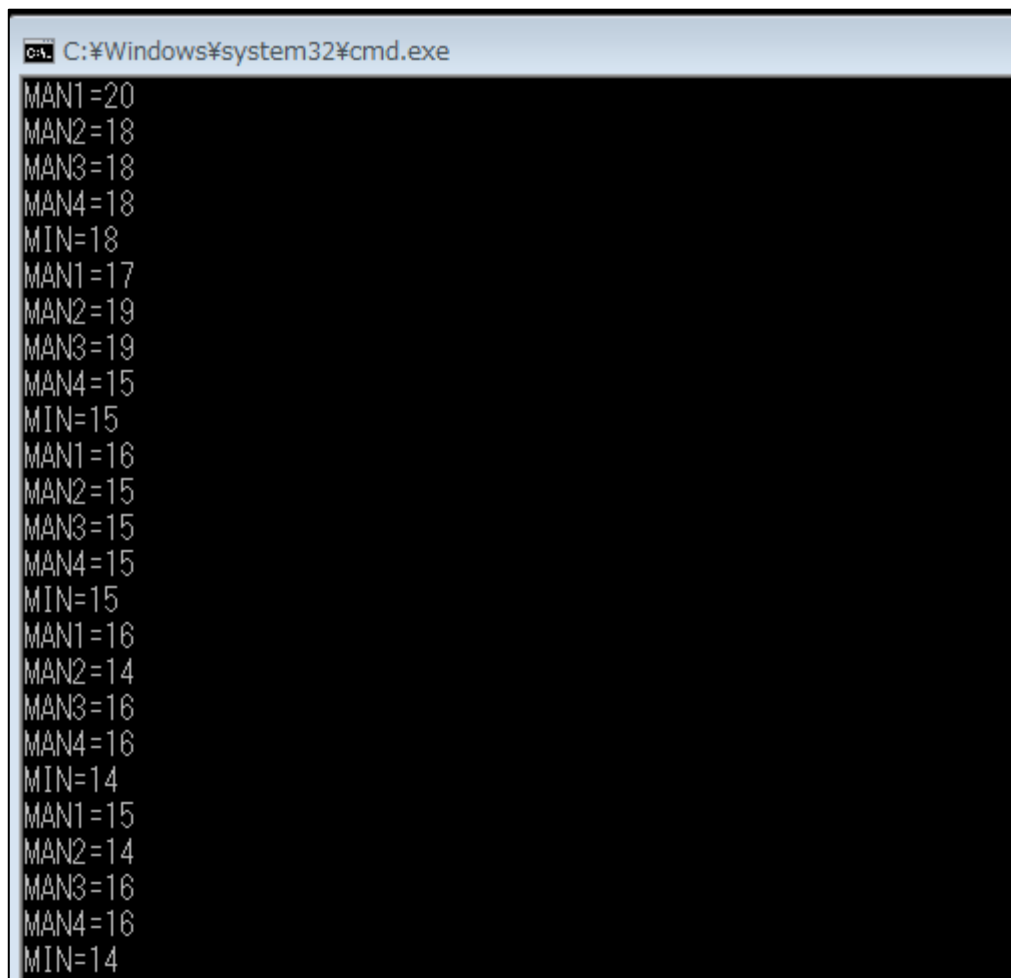
```
317.                                     cell1[x1][y1] = c_3[x1][y1];
318.                                     }
319.                                 }
320.                            }
321.                    if (min == man4) {
322.                        //最小コストをスタックに入れる
323.                        enqueue(min);
324.                        hozon(c_4);
325.                        for (int x1 = 0; x1 <= 2; x1++) {
326.                            for (int y1 = 0; y1 <= 2; y1++) {
327.                                cell1[x1][y1] = c_4[x1][y1];
328.                            }
329.                        }
330.                    }
331.                    //ファイル出力
332.                    FILE *outputfile;
333.                    outputfile = fopen("output.txt", "w");
334.                    for (int x1 = 0; x1 <= 2; x1++) {
335.                        for (int y1 = 0; y1 <= 2; y1++) {
336.                            fprintf(outputfile, "%d ¥n",
                                cell1[x1][y1]);
337.                        }
338.                    }
339.                    fprintf(outputfile, "発見的関数の値=%d ¥n", min);
340.                    fclose(outputfile);
```

3.実行結果

以下に実行結果を示す（解は出ていません）。

実行環境：OS Windows7(64bit)

Visual Studio 2013



```
C:\Windows\system32\cmd.exe
MAN1=20
MAN2=18
MAN3=18
MAN4=18
MIN=18
MAN1=17
MAN2=19
MAN3=19
MAN4=15
MIN=15
MAN1=16
MAN2=15
MAN3=15
MAN4=15
MIN=15
MAN1=16
MAN2=14
MAN3=16
MAN4=16
MIN=14
MAN1=15
MAN2=14
MAN3=16
MAN4=16
MIN=14
```

図 2.実行結果