

人工知能Ⅱ

問題の状態空間表現の課題

5CS16 大和田茜

- アルゴリズムの説明

アルゴリズム説明として以下の図1にフローチャートを示す。実際にプログラムで実装した部分は、青枠で囲われていないところとする。

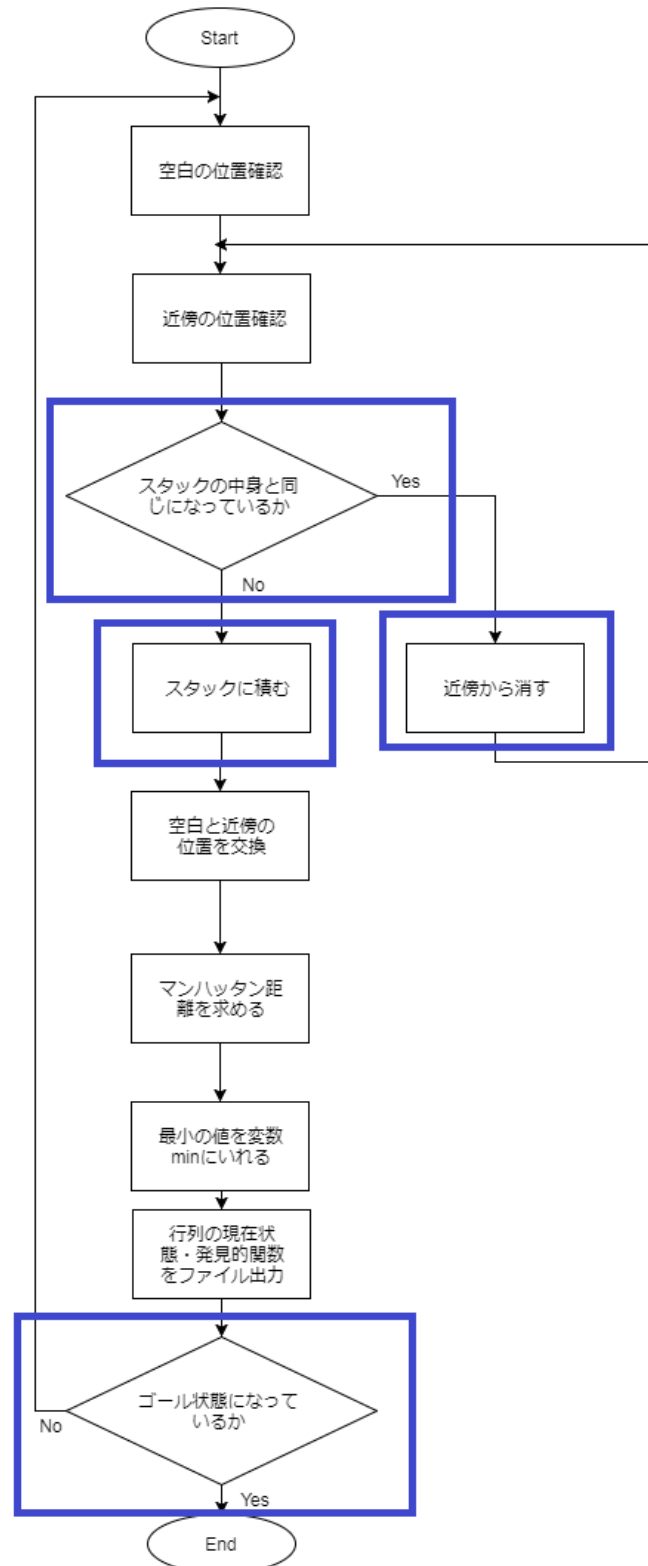
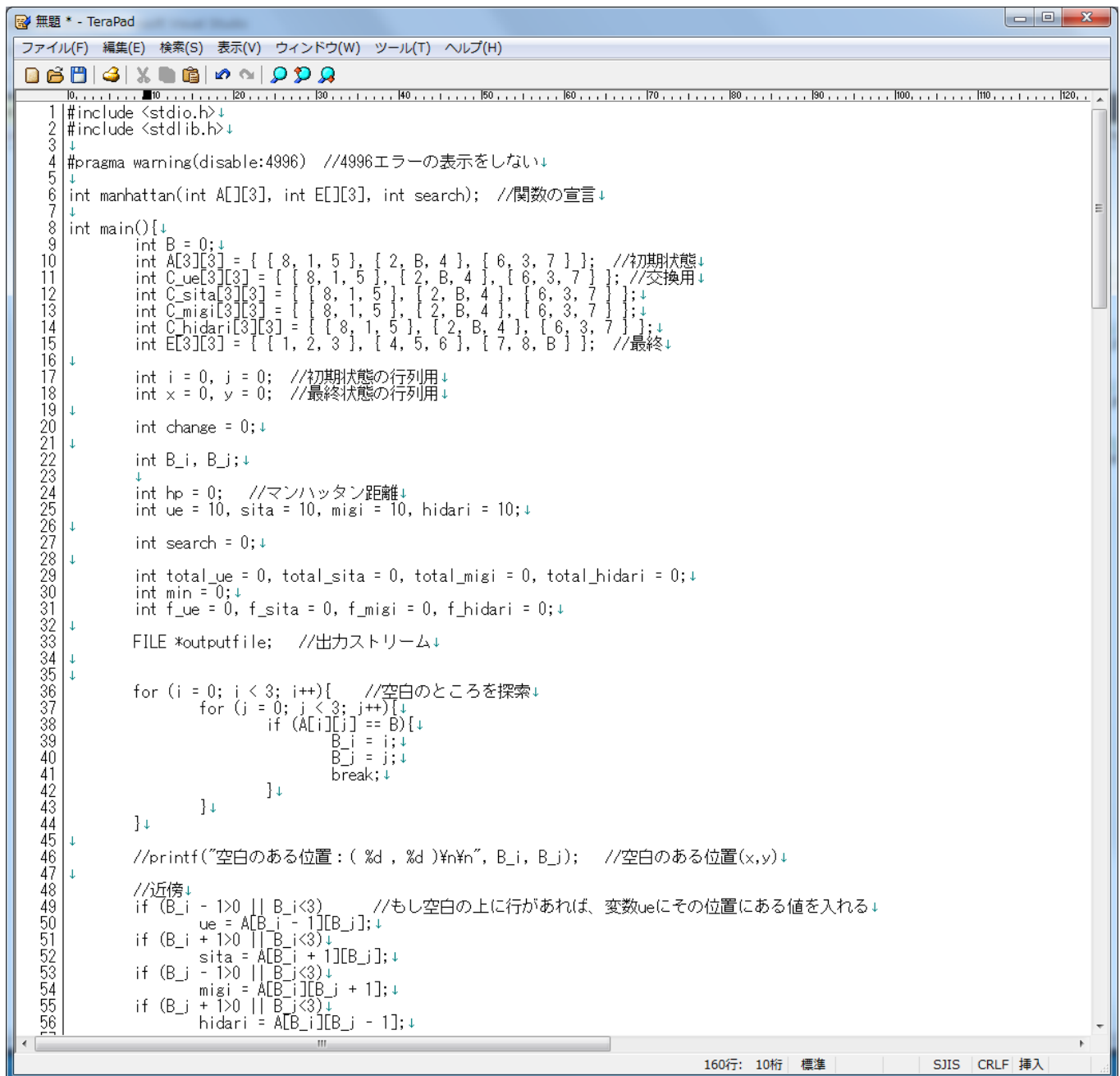


図1．アルゴリズムのフローチャート

- ソースコード

ソースコードを以下の図2から図5に示す。



```

1 #include <stdio.h>↓
2 #include <stdlib.h>↓
3 ↓
4 #pragma warning(disable:4996) //4996エラーの表示をしない↓
5 ↓
6 int manhattan(int A[][3], int E[][3], int search); //関数の宣言↓
7 ↓
8 int main()↓
9     int B = 0;↓
10     int A[3][3] = { { 8, 1, 5 }, { 2, B, 4 }, { 6, 3, 7 } }; //初期状態↓
11     int C_ue[3][3] = { { 8, 1, 5 }, { 2, B, 4 }, { 6, 3, 7 } }; //交換用↓
12     int C_sita[3][3] = { { 8, 1, 5 }, { 2, B, 4 }, { 6, 3, 7 } };↓
13     int C_migi[3][3] = { { 8, 1, 5 }, { 2, B, 4 }, { 6, 3, 7 } };↓
14     int C_hidari[3][3] = { { 8, 1, 5 }, { 2, B, 4 }, { 6, 3, 7 } };↓
15     int E[3][3] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, B } }; //最終↓
16 ↓
17     int i = 0, j = 0; //初期状態の行列用↓
18     int x = 0, y = 0; //最終状態の行列用↓
19 ↓
20     int change = 0;↓
21 ↓
22     int B_i, B_j;↓
23 ↓
24     int hp = 0; //マンハッタン距離↓
25     int ue = 10, sita = 10, migi = 10, hidari = 10;↓
26 ↓
27     int search = 0;↓
28 ↓
29     int total_ue = 0, total_sita = 0, total_migi = 0, total_hidari = 0;↓
30     int min = 0;↓
31     int f_ue = 0, f_sita = 0, f_migi = 0, f_hidari = 0;↓
32 ↓
33     FILE *outputfile; //出力ストリーム↓
34 ↓
35 ↓
36     for (i = 0; i < 3; i++){ //空白の場所を探索↓
37         for (j = 0; j < 3; j++){↓
38             if (A[i][j] == B){↓
39                 B_i = i;↓
40                 B_j = j;↓
41                 break;↓
42             }↓
43         }↓
44     }↓
45 ↓
46     //printf("空白のある位置: ( %d , %d )\n", B_i, B_j); //空白のある位置(x,y)↓
47 ↓
48     //近傍↓
49     if (B_i - 1 > 0 || B_i < 3) //もし空白の上に行があれば、変数ueにその位置にある値を入れる↓
50         ue = A[B_i - 1][B_j];↓
51     if (B_i + 1 > 0 || B_i < 3)↓
52         sita = A[B_i + 1][B_j];↓
53     if (B_j - 1 > 0 || B_j < 3)↓
54         migi = A[B_i][B_j + 1];↓
55     if (B_j + 1 > 0 || B_j < 3)↓
56         hidari = A[B_i][B_j - 1];↓

```

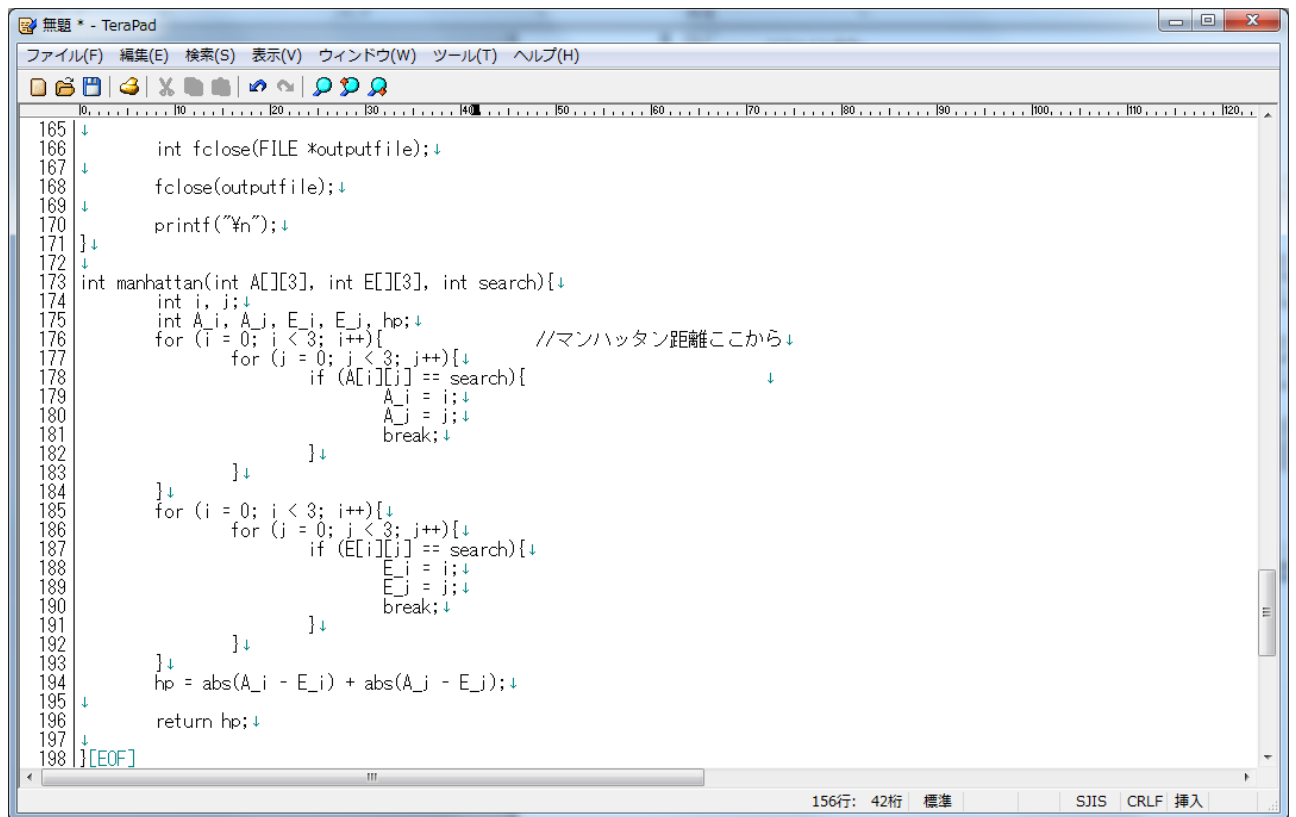
図2 作成プログラム①

```
無題 * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
10 110 120
55 if (B_j + 1 > 0 || B_j < 3){
56     hidari = A[B_i][B_j - 1];
57 }
58 //printf("%d %d\n", ue, sita);
59
60 //交換
61 if (ue < 10){
62     //Bとueの位置を交換
63     change = C_ue[B_i - 1][B_j];
64     C_ue[B_i - 1][B_j] = C_ue[B_i][B_j];
65     C_ue[B_i][B_j] = change;
66 }
67 for (i = 0; i < 9; i++){
68     hp = manhattan(C_ue, E, i);
69     //printf("マンハッタン距離: %d %n", hp);
70     total_ue = total_ue + hp;
71 } //printf("上 合計: %d\n", total_ue);
72 }
73 if (sita < 10){
74     change = C_sita[B_i + 1][B_j];
75     C_sita[B_i + 1][B_j] = C_sita[B_i][B_j];
76     C_sita[B_i][B_j] = change;
77 }
78 for (i = 0; i < 9; i++){
79     hp = manhattan(C_sita, E, i);
80     //printf("マンハッタン距離: %d %n", hp);
81     total_sita = total_sita + hp;
82 } //printf("下 合計: %d\n", total_sita);
83 }
84 if (migi < 10){
85     change = C_migi[B_i][B_j + 1];
86     C_migi[B_i][B_j + 1] = C_migi[B_i][B_j];
87     C_migi[B_i][B_j] = change;
88 }
89 for (i = 0; i < 9; i++){
90     hp = manhattan(C_migi, E, i);
91     //printf("マンハッタン距離: %d %n", hp);
92     total_migi = total_migi + hp;
93 } //printf("右 合計: %d\n", total_migi);
94 }
95 if (hidari < 10){
96     change = C_hidari[B_i][B_j - 1];
97     C_hidari[B_i][B_j - 1] = C_hidari[B_i][B_j];
98     C_hidari[B_i][B_j] = change;
99 }
100 for (i = 0; i < 9; i++){
101     hp = manhattan(C_hidari, E, i);
102     //printf("マンハッタン距離: %d %n", hp);
103     total_hidari = total_hidari + hp;
104 } //printf("左 合計: %d\n", total_hidari);
105 }
106
107 //最小を選択
108 if (total_ue <= total_sita){
109     min = total_ue;
110     f_ue = 1;
111 }
```

図3 作成プログラム②

```
無題 * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
109 min = total_ue;↓
110 f_ue = 1;↓
111 }↓
112 else↓
113 min = total_sita;↓
114 f_sita = 1;↓
115 }↓
116 if (total_migi <= min){↓
117 min = total_migi;↓
118 f_migi = 1;↓
119 f_ue = 0, f_sita = 0;↓
120 }↓
121 if (total_hidari <= min){↓
122 min = total_hidari;↓
123 f_hidari = 1;↓
124 f_ue = 0, f_sita = 0, f_migi = 0;↓
125 }↓
126 printf("最小: %d\n", min);↓
127 ↓
128 //ファイル出力↓
129 outputfile = fopen("d.txt", "w"); //書き込み用↓
130 if (outputfile == NULL){↓
131 printf("cannot open\n");↓
132 exit(1);↓
133 }↓
134 ↓
135 ↓
136 //最小を表示↓
137 if (f_ue == 1){↓
138 //printf("%d %d %d %d %d %d %d %d\n", C_ue[0][0], C_ue[0][1], C_ue[0][2], C_ue[1][0], C_ue[1][1], ↓
139 C_ue[1][2], C_ue[2][0], C_ue[2][1], C_ue[2][2]); //ターミナル表示↓
140 fprintf(outputfile, "%d %d %d %d %d %d %d %d\n", C_ue[0][0], C_ue[0][1], C_ue[0][2], C_ue[1][0], ↓
141 C_ue[1][1], C_ue[1][2], C_ue[2][0], C_ue[2][1], C_ue[2][2]); //ファイルに書く↓
142 }↓
143 if (f_sita == 1){↓
144 //printf("%d %d %d %d %d %d %d %d\n", C_sita[0][0], C_sita[0][1], C_sita[0][2], C_sita[1][0], ↓
145 C_sita[1][1], C_sita[1][2], C_sita[2][0], C_sita[2][1], C_sita[2][2]);↓
146 fprintf(outputfile, "%d %d %d %d %d %d %d %d\n", C_sita[0][0], C_sita[0][1], C_sita[0][2], ↓
147 C_sita[1][0], C_sita[1][1], C_sita[1][2], C_sita[2][0], C_sita[2][1], C_sita[2][2]);↓
148 }↓
149 if (f_migi == 1){↓
150 //printf("%d %d %d %d %d %d %d %d\n", C_migi[0][0], C_migi[0][1], C_migi[0][2], C_migi[1][0], ↓
151 C_migi[1][1], C_migi[1][2], C_migi[2][0], C_migi[2][1], C_migi[2][2]);↓
152 fprintf(outputfile, "%d %d %d %d %d %d %d %d\n", C_migi[0][0], C_migi[0][1], C_migi[0][2], ↓
153 C_migi[1][0], C_migi[1][1], C_migi[1][2], C_migi[2][0], C_migi[2][1], C_migi[2][2]);↓
154 }↓
155 if (f_hidari == 1){↓
156 //printf("%d %d %d %d %d %d %d %d\n", C_hidari[0][0], C_hidari[0][1], C_hidari[0][2], C_hidari[1][0], ↓
157 C_hidari[1][1], C_hidari[1][2], C_hidari[2][0], C_hidari[2][1], C_hidari[2][2]);↓
158 fprintf(outputfile, "%d %d %d %d %d %d %d %d\n", C_hidari[0][0], C_hidari[0][1], C_hidari[0][2], ↓
159 C_hidari[1][0], C_hidari[1][1], C_hidari[1][2], C_hidari[2][0], C_hidari[2][1], C_hidari[2][2]);↓
160 }↓
161 printf("発見の関数: %d\n", 1 + min);↓
162 ↓
163 fprintf(outputfile, "発見の関数: %d\n", 1 + min);↓
164 ↓
104行: 42桁 標準 SJIS CRLF 挿入
```

図4 作成プログラム③



```
165  
166     int fclose(FILE *outputfile);  
167  
168     fclose(outputfile);  
169  
170     printf("%n");  
171 }  
172  
173 int manhattan(int A[][3], int E[][3], int search){  
174     int i, j;  
175     int A_i, A_j, E_i, E_j, hp;  
176     for (i = 0; i < 3; i++){ //マンハッタン距離ここから  
177         for (j = 0; j < 3; j++){  
178             if (A[i][j] == search){  
179                 A_i = i;  
180                 A_j = j;  
181                 break;  
182             }  
183         }  
184     }  
185     for (i = 0; i < 3; i++){  
186         for (j = 0; j < 3; j++){  
187             if (E[i][j] == search){  
188                 E_i = i;  
189                 E_j = j;  
190                 break;  
191             }  
192         }  
193     }  
194     hp = abs(A_i - E_i) + abs(A_j - E_j);  
195  
196     return hp;  
197 }  
198 ][EOF]
```

図5 作成プログラム④

● プログラムの実行結果

✓ 動作環境

- Windows 7 Professional ver.6.1
- Visual Studio 2013
- C++

✓ 実行結果

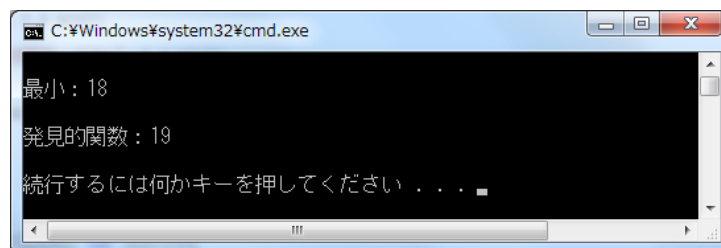


図6 デバック画面

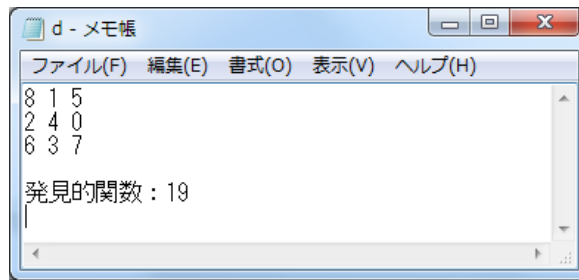


図 7 出力ファイル

✓ 発見的関数の推移グラフ

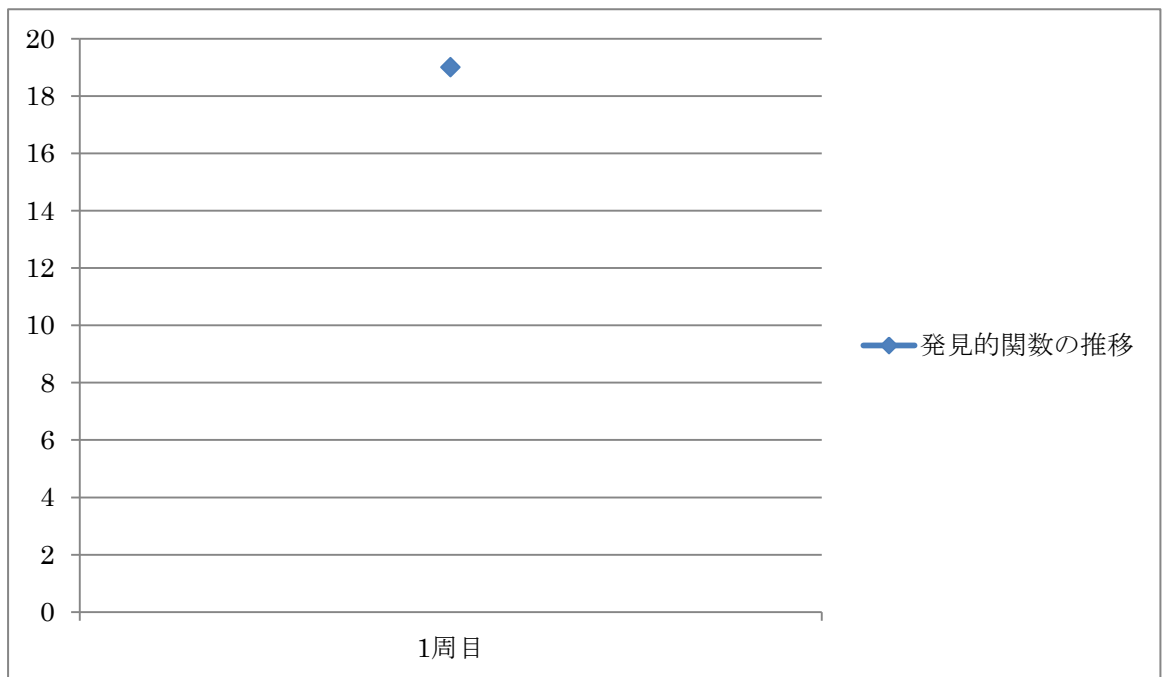


図 8 発見的関数の推移