

1 相互作用するマルチエージェント

マルチエージェントシミュレーションでは、エージェント同士の相互作用が重要です。そこで、sa1.c を改造して、エージェントと他のエージェントとの相互作用のシミュレーションを行いましょう。

具体的には、エージェントを2種類に分けることにします。ひとつはsa1.c プログラムでシミュレートしたような、ランダムウォークを行うエージェント Ar です。もうひとつは、平面の右上に向けて直進する、sa0.c プログラムでシミュレートしたような単純なエージェント As です。この2種類のエージェントを同じ平面上で動作させます。

両者の相互作用として、ランダムウォークエージェント Ar が直進エージェント As と一定距離以内に接近すると、Ar は、ランダムウォークをやめて、As と同じように直進し始めることにします。つまり、Ar が As に近づくと、Ar が As に変身してしまいます。

Cによる数値計算とシミュレーション 小高 知宏 オーム社 170 頁より

2 相互作用するマルチエージェント sa2.c

```
/*
   sa2.c
   シンプルなエージェントシミュレーション
   2次元平面内で動作するエージェント群
   2種類のエージェントが相互作用します。
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define BUFSIZE 256;
#define N 30
#define N_OF_A 2
#define TIMELIMIT 100
#define SEED RAND_MAX-1
#define R 0.1
#define DX 0.1
#define DY 0.1

struct coordinate{
    double x;
    double y;
};

struct agent{
```

```

    int category;
    struct coordinate coord;
    double dattribute[N_OF_A];
    int iattribute[N_OF_A];
};

void  calcnext(struct agent a[]);
void  putstate(struct agent a[], int t);
double frand(void);
void  cat0(struct agent *cat0agent, struct agent a[]);
void  cat1(struct agent *cat0agent);

int main()
{
    struct agent a[N] = {0};
    int    t;

    srand(SEED);
    a[0].category = 1;
    a[0].coord.x = -2;
    a[0].coord.y = -2;

    putstate(a, 0);

    for(t = 1; t <= TIMELIMIT; ++t){
        calcnext(a);
        putstate(a, t);
    }

    return 0;
}

void calcnext(struct agent a[])
{
    int i;
    for(i = 0; i < N; ++i){
        if(a[i].category == 0){
            cat0(&a[i], a);
        }else{
            if(a[i].category == 1){
                cat1(&a[i]);
            }else{
                fprintf(stderr, "ERROR カテゴリがありません (%d)\n", i);
            }
        }
    }
}

void cat0(struct agent *cat0agent, struct agent a[])

```

```

{
    int i;
    double c0x,c0y,ax,ay;

    for(i = 0; i < N; ++i){
        if(a[i].category == 1){
            c0x = (*cat0agent).coord.x;
            c0y = (*cat0agent).coord.y;
            ax = a[i].coord.x;
            ay = a[i].coord.y;
            if(((c0x - ax) * (c0x - ax) + (c0y - ay) * (c0y - ay)) < R){
                (*cat0agent).category = 1;
                return;
            }
        }
    }

    (*cat0agent).coord.x += (frand() - 0.5);
    (*cat0agent).coord.y += (frand() - 0.5);
}

void cat1(struct agent *cat1agent)
{
    (*cat1agent).coord.x += DX;
    (*cat1agent).coord.y += DX;
}

void putstate(struct agent a[], int t)
{
    int i;
    printf("t = %d\n", t);
    for(i = 0; i < N; ++i){
        printf("%d\t%lf\t%lf\n", a[i].category,
                a[i].coord.x,a[i].coord.y);
    }
}

double frand(void)
{
    return (double)rand()/RAND_MAX;
}

```

3 「sa2」の操作方法

「sa2」の操作方法

./sa2

上記の「./」は、Linux でのプログラムを実行する時の例です。
ウィンドウズでは、sa2 「Enter」と入力してください。

4 「sa2」の実行結果

```
t = 100
1 8.000000 8.000000
0 3.143406 0.465987
0 0.228126 6.522355
0 -2.416883 -1.487838
0 -0.442244 4.416120
1 8.468151 8.003171
0 0.170629 2.157018
1 8.713367 8.021696
0 -0.177801 2.891568
1 7.951296 8.147011
0 2.591723 -5.928404
1 7.483996 7.659264
1 8.409720 7.997079
0 0.437574 0.725578
0 -1.376568 3.002869
1 7.842363 7.780593
0 4.042177 -0.142888
1 8.107361 7.980007
0 0.075076 2.661977
1 7.971938 8.155690
1 8.162227 8.029624
1 8.402725 7.831541
0 -3.917692 2.092898
1 7.825187 7.944819
0 4.148015 1.210181
0 -1.162999 -6.175085
0 -3.667287 1.151555
1 7.879473 8.220090
0 -2.523331 -3.355205
1 8.289355 8.067730
```

5 マルチエージェントへの拡張 gsa2.c

gnuplot へ出力します。

```
/*
   gsa2.c
   シンプルなエージェントシミュレーション
   2次元平面内で動作するエージェント
   2種類のエージェントが相互作用します
   gnuplot に出力します。
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>

#define BUFSIZE 256;
#define N 50
#define N_OF_A 2
#define TIMELIMIT 100
#define SEED RAND_MAX-1
#define R 0.1
#define DX 0.1
#define DY 0.1

#define ATMPFILE "atmpfile.tmp"
#define BTMPFILE "btempfile.tmp"
#define GNUPLOT "c:\\gnuplot\\binary\\gnuplot"
#define RANGE 10

struct coordinate{
    double x;
    double y;
};

struct agent{
    int category;
    struct coordinate coord;
    double dattribute[N_OF_A];
    int iattribute[N_OF_A];
};

void calcnext(struct agent a[]);
void fputstate(struct agent a[], int t);
double frand(void);
void cat0(struct agent *cat0agent, struct agent a[]);
void cat1(struct agent *cat0agent);

int main()
```

```

{
    struct agent a[N] = {0};
    int t;
    FILE *pipe;

    if((pipe = popen(GNUPLOT " -persist", "w")) == NULL){
        fprintf(stderr, "パイプが開けません。 \n");
        exit(1);
    }

    fprintf(pipe, "set xrange [-%d:%d]\n", RANGE, RANGE);
    fprintf(pipe, "set yrange [-%d:%d]\n", RANGE, RANGE);

    srand(SEED);
    a[0].category = 1;
    a[0].coord.x = -2;
    a[0].coord.y = -2;

    for(t = 1; t <= TIMELIMIT; ++t){
        printf("t = %d\n", t);
        calcnext(a);
        fputstate(a, t);
        fprintf(pipe,
            "plot \"\" ATMPFILE \"\" , \"\" BTMPFILE \"\" with points pt 4 \n");
        fflush(pipe);
        usleep(100000);
    }
    getchar();
    return 0;
}

void calcnext(struct agent a[])
{
    int i;
    for(i = 0; i < N; ++i){
        if(a[i].category == 0){
            cat0(&a[i], a);
        }else{
            if(a[i].category == 1){
                cat1(&a[i]);
            }else{
                fprintf(stderr, "ERROR カテゴリがありません (%d)\n", i);
            }
        }
    }
}

void cat0(struct agent *cat0agent, struct agent a[])
{

```

```

int i;
double c0x,c0y,ax,ay;

for(i = 0; i < N; ++i){
    if(a[i].category == 1){
        c0x = (*cat0agent).coord.x;
        c0y = (*cat0agent).coord.y;
        ax = a[i].coord.x;
        ay = a[i].coord.y;
        if(((c0x - ax) * (c0x - ax) + (c0y - ay) * (c0y - ay)) < R){
            (*cat0agent).category = 1;
            return;
        }
    }
}

(*cat0agent).coord.x += (frand() - 0.5);
(*cat0agent).coord.y += (frand() - 0.5);
}

void cat1(struct agent *cat1agent)
{
    (*cat1agent).coord.x += DX;
    (*cat1agent).coord.y += DX;
}

void fputstate(struct agent a[], int t)
{
    int i;
    FILE *fpa, *fpb;

    if((fpa = fopen(ATMPFILE, "w")) == NULL){
        fprintf(stderr, "一時ファイルAが開けません\n");
        exit(1);
    }

    if((fpb = fopen(BTMPFILE, "w")) == NULL){
        fprintf(stderr, "一時ファイルBが開けません\n");
        exit(1);
    }

    for(i = 0; i < N; ++i){
        if(a[i].category == 0){
            fprintf(fpa, "%lf %lf\n", a[i].coord.x, a[i].coord.y);
        }else{
            fprintf(fpb, "%lf %lf\n", a[i].coord.x, a[i].coord.y);
        }
    }
    fclose(fpa);
}

```

```
    fclose(fpb);  
}  
  
double frand(void)  
{  
    return (double)rand()/RAND_MAX;  
}
```

6 「gsa2」の操作方法

「gsa2」の操作方法

```
./gsa2
```

上記の「./」は、Linuxでのプログラムを実行する時の例です。
ウィンドウズでは、gsa2 「Enter」と入力してください。

7 「gsa2」の実行結果

別紙を参照してください。