1　交叉　遺伝子の適応度計算確認

　前回は、遺伝子の適応度の計算確認をしました。結果とプログラム、第 0 世代の遺伝子群は、下記の通りです。

2　結果

番号	遺伝子	適応度
0	00000110110111101010	12
1	11000110100010100110	61
2	11110011111011011001	82
3	00110001101000111010	-26
4	00111011111110010000	181
5	11101101011111000101	-8
6	11000100000110001010	-10
7	01110011001001100011	6
8	11110001110011100111	-12
9	01100101100111010110	-130
10	01011011010011010111	-43
11	11101000100001110010	-39
12	00101100110000010110	53
13	01110100101100100110	-75
14	10100010100011000101	-91
15	11100000000010111100	-114
16	01111100001000110000	53
17	00011010001100001000	48
18	11011001010010001100	205
19	00110011101101100010	21
20	00010100000100111010	-145
21	01100001100101000011	-69
22	10110001011101011110	-142
23	01101101101001010011	-51
24	11100111001111110011	-123
25	01101101001001001010	116
26	11011110010110001001	194
27	10010110111100110011	-67
28	11101111001001101010	217
29	01001010000001011110	-118

3　ソースプログラム

```
/*
適応度の計算
ファイルから読み込みます。
fevaifit.c
*/
```

```
#include <stdio.h>
#include <stdlib.h>
#define DATANO 20

int main()
{
  int q[] = {31, 41, 59, 26, 53, 58, 97, 93, 23, 84,
            -62,-64,-33,-83,-27,-95,-2,-88,-41,-97};

  FILE *fp;
  char buff[21];
  int number;
  int fitness = 0;
  int i;
  int j;

  if((fp = fopen("gene.txt", "r")) == NULL){
    printf("ファイルを開くことが出来ません。");
    exit(1);
  }

  j = 0;
  while(fgets(buff, 21, fp) != NULL){
    if((j % 2) == 0){                  //読み飛ばしています。

      fitness = 0;
      for(i = 0; i < 20; i++){
        fitness += (buff[i] - 48) * q[i];
      }
      printf("%4d %s %4d\n", j/2, buff, fitness);
    }
    j++;
  }
  fclose(fp);
  return 0;
}
```

4 第0世代の遺伝子群

```
00000110110111101010
11000110100010100110
11110011111011011001
00110001101000111010
00111011111110010000
11101101011111000101
11000100000110001010
01110011001001100011
```

```
11110001110011100111
01100101100111010110
01011011010011010111
11101000100001110010
00101100110000010110
01110100101100100110
10100010100011000101
11100000000010111100
01111100001000110000
00011010001100001000
11011001010010001100
00110011101101100010
00010100000100111010
01100001100101000011
10110001011101011110
01101101101001010011
11100111001111110011
01101101001001001010
11011110010110001001
10010110111100110011
11101111001001101010
01001010000001011110
```

この後プログラムは、各遺伝子ごとの適応度を計算して、ルーレット配列に適応度に保管します。
そして、遺伝子の世代全体の適応度として、ルーレット配列に保管された適応度を合算します。
ルーレット配列の値と世代の適応度を使って、一組の親遺伝子を選び、一点交叉で一組の子遺伝を作ります。
一点交叉で一組の親遺伝子から子遺伝子が作られているかを調べます。

5 一点交叉

交叉させる親固体 ia, ib を選択して交叉位置をランダムに決定し、その位置の後で遺伝子情報を入れ替えて子固体 ia, ib を作成いる。

親個体 1a [1,2,3,4,5]
親個体 1b [6,7,8,9,10]

交叉位置を 3 番目とすると

子個体 2a [1,2,3,9,10]
子個体 2b [6,7,8,4,5]

6 遺伝的アルゴリズム　sga.c

```
/*
  sga.c
  sga の説明用プログラム
  使い方
  ./sga
*/

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

#define DATANO 20
#define POOLSIZE 30
#define LASTG 200
#define MRATE 0.01
#define SEED 65535
#define YES 1
#define NO 0
#define VALUE 499
#define MAXFIT 1000

int evalfit(int gene[]);
void mating(int pool[POOLSIZE][DATANO]);
void mutation(int pool[POOLSIZE][DATANO]);
void printp(int pool[POOLSIZE][DATANO]);
void initpool(int pool[POOLSIZE][DATANO]);

int rndn(int l);
int notval(int v);
int select(int roulette[POOLSIZE], int totalfitness);

void copypool(int pool[POOLSIZE][DATANO], int nextpool[POOLSIZE][DATANO]);
void crossing(int m[], int p[], int c1[], int c2[]);

int q[] = {31, 41, 59, 26, 53, 58, 97, 93, 23, 84,
           -62,-64,-33,-83,-27,-95,-2,-88,-41,-97};

int main(int argc, char *argv[])
{
  int pool[POOLSIZE][DATANO];
  int generation;

  srand(SEED);

  initpool(pool);

  for(generation=0;generation<LASTG;++generation){  //2013/10/7
```

```
    printf("%d 世代\n", generation);
    mating(pool);
    mutation(pool);
    printp(pool);
  }

  return 0;
}


void mating(int pool[POOLSIZE][DATANO])
{
  int i;
  int totalfitness = 0;
  int nextpool[POOLSIZE][DATANO];  // POOLSIZE=30 DATANO=20
  int roulette[POOLSIZE];
  int mama,papa;

  for(i = 0; i < POOLSIZE; ++i){
    roulette[i] = evalfit(pool[i]);
    totalfitness += roulette[i];
  }

  for(i = 0; i < POOLSIZE/2; ++i){
    do{
        mama = select(roulette, totalfitness);
        papa = select(roulette, totalfitness);
    }while(mama == papa);

    crossing(pool[mama], pool[papa], nextpool[i*2], nextpool[i*2+1]);
  }

  copypool(pool, nextpool);
}


int select(int roulette[POOLSIZE], int totalfitness)
{
  int i;
  int ball;
  int acc = 0;  //2013/10/7

  ball = rndn(totalfitness);
  for(i = 0; i < POOLSIZE; ++i){
    acc += roulette[i];
    if(acc > ball)break;
  }

  return i;
```

```
}

void crossing(int m[], int p[], int c1[], int c2[])
{
  int j;
  int cp;

  cp = rndn(DATANO);  // DATANO = 20

  for(j = 0; j < cp; ++j){
    c1[j] = m[j];
    c2[j] = p[j];
  }

  for(; j < DATANO; ++j){
    c2[j] = m[j];
    c1[j] = p[j];
  }
}

void copypool(int pool[POOLSIZE][DATANO], int nextpool[POOLSIZE][DATANO])
{
  int i,j;

  for(i = 0; i < POOLSIZE; ++i){
    for(j = 0; j < DATANO; ++j){
      pool[i][j] = nextpool[i][j];
    }
  }
}

int evalfit(int g[])
{
  int i;
  int fitness = 0;

  for(i = 0; i < DATANO; ++i){
    fitness += g[i] * q[i];
  }

  return MAXFIT - abs(VALUE - fitness);
}

void printp(int pool[POOLSIZE][DATANO])
{
  int i,j;
  int fitness;
  double totalfitness = 0;
  int elite, bestfit = 0;
```

```
  for(i = 0; i < POOLSIZE; ++i){
    for(j = 0; j < DATANO; ++j){
      printf("%ld", pool[i][j]);
    }
    fitness = evalfit(pool[i]);
    printf("\t%d\n", fitness);
    if(fitness > bestfit){
      bestfit = fitness;
      elite = i;
    }
    totalfitness += fitness;
  }

  printf("%d\t%d  ", elite, bestfit);

  printf("%lf\n", totalfitness / POOLSIZE);
}

void initpool(int pool[POOLSIZE][DATANO])
{
  int i,j;

  for(i = 0; i < POOLSIZE; ++i){
    for(j = 0; j < DATANO; ++j){
      pool[i][j] = rndn(2);
    }
  }
}

int rndn(int l)
{
  int rndno;

  while((rndno = ((double)rand() / RAND_MAX) * l) == l);

  return rndno;
}

void mutation(int pool[POOLSIZE][DATANO])
{
  int i,j;

  for(i = 0; i < POOLSIZE; ++i){
    for(j = 0; j < DATANO; ++j){
      if((double)rndn(100)/ 100.0 <= MRATE){
        pool[i][j] = notval(pool[i][j]);
      }
    }
```

```
  }
}

int notval(int v)
{
  if(v == YES){
    return NO;
  }else{
    return YES;
  }
}
```

7 練習問題 51
rensyu51.c 「C 言語」233 頁より

ファイルの内容を 1 文字単位で別のファイルにコピーする COPY コマンドを作りなさい。

```
#include <stdio.h>
#include <process.h> /*Windows 環境でコンパイルするとき*/
/*#include <stdlib.h>  Linux 環境でコンパイルするとき*/

int main(int argc, char *argv[])
{
  int  c;
  FILE *fpi,*fpo;

  if(argc != 3){
    printf("Arg is not correct\n");
    exit(1);
  }

  if((fpi = fopen(argv[1], /*????*/)) == NULL){
    printf("Can not open File\n");
    exit(1);
  }

  if((fpo = fopen(/*????*/, /*????*/)) == NULL){
    printf("Can not create File\n");
    exit(1);
  }

  while((/*????*/) != EOF){
    /*????*/;
  }

  fclose(fpi);
```

```
  fclose(fpo);

  return 0;
}
```