

```

1  //sciara_pga.c
2
3  #include <stdio.h>
4  #include <pgapack.h>
5  #include <sys/types.h>
6  #include <unistd.h>
7  #include <time.h>
8
9  #define PAR_NUM      8
10 #define IND_LEN      64
11 #define POPSIZE      16
12 #define ITERATIONS   100
13
14 int  nbits[PAR_NUM] = {8,      8,      8,      8,      8,      8,      8,      8 };
15 float low  [PAR_NUM] = {0.01, 0.4,  2.0,  1095.0, 1000.0, 800.0, 1.0e-16, 0.001};
16 float high [PAR_NUM] = {0.3,  1.0, 10.0, 1150.0, 1094.0, 900.0, 1.0e-11, 1.0};
17
18 double sciaraEvaluationFunction(PGAContext *, int, int);
19
20 /*****
21  *          user main program
22  *****/
23 int main( int argc, char **argv ) {
24     PGAContext *ctx;
25     time_t start_t, end_t;
26     double diff_t;
27
28     start_t = time(NULL);
29
30     ctx = PGACreate(&argc, argv, PGA_DATATYPE_BINARY, IND_LEN, PGA_MAXIMIZE);
31     PGASetRandomSeed(ctx, 1);
32
33     PGASetPopSize(ctx, POPSIZE);
34     PGASetMaxGAIterValue(ctx, ITERATIONS);
35     PGASetNumReplaceValue(ctx, POPSIZE / 2);
36     PGASetPopReplaceType(ctx, PGA_POPREPL_BEST);
37     PGASetPrintFrequencyValue(ctx, 1);
38
39     PGASetUp(ctx);
40     PGARun(ctx, sciaraEvaluationFunction);
41     PGADestroy(ctx);
42
43     end_t = time(NULL);
44     diff_t = difftime(end_t, start_t);
45     printf("Elapsed time = %f\n", diff_t);
46
47     return(0);
48 }
49
50 /*****
51  *  user defined evaluation function
52  *  ctx - contex variable
53  *  p   - chromosome index in population
54  *  pop - which population to refer to
55  *****/
56 int sciara_exec(char* program, char** arg_list)
57 {
58     pid_t child_pid;
59     child_pid = fork();
60     if (child_pid != 0)
61         return child_pid;
62     else
63     {
64         execvp(program, arg_list);
65         fprintf(stderr, "An error occurred. Program terminated.\n");
66         abort();
67     }
68 }
69
70 double sciaraEvaluationFunction(PGAContext *ctx, int p, int pop) {
71
72     FILE *f;
73     char parameter_path[] = "../sciara/param.txt",
74          fitness_path[]   = "../sciara/fitness.txt",
75          *arg_list[] = {
76         "../sciara/sciara.sh",

```

```
77         NULL
78     },
79     str[256];
80
81     int child_status;
82     int i, start = -1, end;
83     float prm[PAR_NUM], e1;
84
85     //set parameters from individual
86     for (i=0; i<PAR_NUM; i++)
87     {
88         start += 1;
89         end = start + nbits[i] - 1;
90         prm[i] = PGAGetRealFromBinary(ctx, p, pop, start, end, low[i], high[i]);
91     }
92
93     //write parameters values on file
94     f=fopen(parameter_path, "w");
95     for (i=0; i<PAR_NUM; i++)
96         if (i==6)
97             fprintf(f, "prm[%d]\t%e\n", i, prm[i]);
98         else
99             fprintf(f, "prm[%d]\t%f\n", i, prm[i]);
100     fclose(f);
101
102     //sciara batch execution
103     sciara_exec(arg_list[0], arg_list);
104     wait(&child_status);
105
106     //read fitness from file
107     f=fopen(fitness_path, "r");
108     fscanf(f, "%s", str);
109     e1 = atof(str);
110
111     //return fitness
112     return(e1);
113 }
114
115
```