

```

1 //FOLDER: parallel_example_explicit
2
3 /*
4  * This is an example program for PGAPack. The objective is to maximize the
5  * function  $y=x^2$  in  $[0, 2^{indlen} - 1]$ .
6  */
7
8 #include <pgapack.h>
9 #include <stdio.h>
10
11 double EvaluationFunction(PGAContext *, int, int);
12
13 /*****
14  *      main program
15  *****/
16 int main( int argc, char **argv ) {
17     PGAContext *ctx;
18     int np, myid, indlen;
19
20     MPI_Init(&argc, &argv);
21     MPI_Comm_size(MPI_COMM_WORLD, &np);
22     MPI_Comm_rank(MPI_COMM_WORLD, &myid);
23
24     if (myid == 0)
25     {
26         printf("String length = ");
27         scanf("%d", &indlen);
28     }
29     MPI_Bcast(&indlen, 1, MPI_INT, 0, MPI_COMM_WORLD);
30
31     ctx = PGACreate(&argc, argv, PGA_DATATYPE_BINARY, indlen, PGA_MAXIMIZE);
32     PGASetPopSize(ctx, 20);
33     PGASetMaxGAIterValue(ctx, 100);
34     PGASetPrintFrequencyValue(ctx, 1);
35     PGASetRandomSeed(ctx, 1);
36     PGASetUp(ctx);
37
38     //PGARun(ctx, EvaluationFunction);
39     PGAEvaluate(ctx, PGA_OLDPOP, EvaluationFunction, MPI_COMM_WORLD);
40     if (myid == 0)
41         PGAFitness(ctx, PGA_OLDPOP);
42     while (!PGADone(ctx, MPI_COMM_WORLD)){
43         if (myid == 0){
44             PGASelect(ctx, PGA_OLDPOP);
45             PGARunMutationAndCrossover(ctx, PGA_OLDPOP, PGA_NEWPOP);
46         }
47         PGAEvaluate(ctx, PGA_OLDPOP, EvaluationFunction, MPI_COMM_WORLD);
48         if (myid == 0)
49             PGAFitness(ctx, PGA_NEWPOP);
50         PGAUpdateGeneration(ctx, MPI_COMM_WORLD);
51         if (myid == 0)
52             PGAPrintReport(ctx, stdout, PGA_OLDPOP);
53     }
54
55     PGADestroy(ctx);
56     MPI_Finalize();
57
58     return(0);
59 }
60
61 /*****
62  *      user defined evaluation function
63  *      ctx - contex variable
64  *      p - chromosome index in population
65  *      pop - which population to refer to
66  *****/
67 double EvaluationFunction(PGAContext *ctx, int p, int pop) {
68     int int_val, stringlen;
69
70     stringlen = PGAGetStringLength(ctx);
71     int_val = PGAGetIntegerFromBinary(ctx, p, pop, 0, stringlen-1);
72
73     return((double) int_val*int_val);
74 }
75
76

```

