Appendix G

GA-SA Program

Program Description

Input:

- The generating unit system data which consists of the heat input-output curves, ramp rate limits, unit operating limits, fuel cost of each unit, initial power outputs at time t=0, forecast load demands, and transmission loss B-matrix coefficients.

Output:

- The result of the program consists of the power output of each generating units and their highest and lowest possible power output, the power loss, the generator fuel cost and the CPU time of each time period.

```
/*-------------------------------------------------------------------------------------
-*/
/* Program : Genetic Algorithm based on Simulated Annealing solutions for Constrained EDP
/* Compile : gcc gasa.c -o GASA -lm
/* Execute : GASA
/*-------------------------------------------------------------------------------------
-*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/* Include Utility Procedure And Functions */
#include "os.h"
#include "gasa.h"
#include "utility.c"
#include "random.c"
#include "interfac.c"
#include "stats.c"
#include "memory.c"
#include "initial.c"
#include "ureport.c"
#include "utriopn.c"
#include "ugeneran.gasa"
#include "samodule.c"

int main()
{
        FILE     *Summary;
        char     SummaryName[30];
        double   SummaryCost;
        Word      i,ref;
        double  P_temp[MaxUnit];
        /* Definds Parameters of Command */

        printf("\n Data Filename(w/o Ext)?");       scanf("%s",InpFileName);
        sprintf(SummaryName,"SummaryOf%s.txt",InpFileName);
        if((Summary = fopen(SummaryName,"w"))==NULL){printf("Summary File error\n");exit(-
1);}

        /* Input GA Parameters */
        EncodeType=1;
        ReadData();
        system(cls);

        printf("\n Input PopSize       :56    ----->");fscanf(InpPara,"%i",&PopSize);
        printf("\n Input MaxGeneration :1500  ----->");fscanf(InpPara,"%i",&MaxGeneration);
        printf("\n Input Epoch         :10    ----->");fscanf(InpPara,"%i",&Epoch);
        printf("\n First mseed         :1     ----->");fscanf(InpPara,"%i",&First_mseed);
        printf("\n Last  mseed         :20    ----->");fscanf(InpPara,"%i",&Last_mseed);
        printf("\nTrial  :1000       ----->");scanf("%d",&TrialNo);
        printf("\nSigma1 : 100       ----->");scanf("%lf",&sigma1);
        printf("\nSigmaf :   1       ----->");scanf("%lf",&sigmaf);

        BestFile = First_mseed;
        system(cls);
        BuildCoef();

        /* Build Fitness Function */
        for (i=0; i < NUnit; i++) P[i]=Pi[i][min]; CostMin=CostFunc(P);
        for (i=0; i < NUnit; i++) P[i]=Pi[i][max]; CostMax=CostFunc(P);
        Diff=1/(CostMax-CostMin);

        /* Check Size of String */
        NGenes = NUnit-1;
        LChrom = ParaBit*NGenes;
        if(LChrom > MaxString)
        {
              printf("Too many Parameter or LChrom > Maxstring\n");exit(-1);
         }

        initmalloc_sga();
        SummaryCost = 0;
        for (mseed=First_mseed; mseed <= Last_mseed; mseed++)
         {
/* Move to inside SA funct srand(mseed);*/
```

```
TotalCost=0;    SumTime.tv_sec = 0;    SumTime.tv_usec = 0;

 /* Make LoopFile */
 sprintf(FileName,"%s%sGASA%i%s",MyFolder,pth,mseed,InpFileName);


 if ((outfp = fopen( FileName, "w"))== NULL)
{   fprintf(stderr, "Cannot open Report file.\n\a");
            exit(-1);
}

fprintf(outfp,"\nGASA result %d Unit Seed No. %d",NUnit,mseed);

 if (ParaBit == 8)  NumSpace=0xFF;
 if (ParaBit == 16) NumSpace=0xFFFF;

  /* Initialize GA Parameters */

  Species[0].PCross=0.5;       Species[0].PMutation=0.01;
  Species[0].ipick=PopSize-1; Species[0].SumFitness=0.0;
  Species[0].RateBias=0.50;
  SeedRandom=mseed*0.05;

  /* Do Process */
  /* Initial Power Generation */
  for (i=0; i < NUnit;  i++) P[i]=Pi[i][min];
  for (i=0; i < NGenes; i++) Empty[i]=0;
  for (i=0; i < NGenes; i++) Full[i] =~0x0;
  TitleReport();

  for (Hour=0; Hour < HourDay; Hour++)
  {
       Generation=0;      NMigrate=0;
       TerminateMode=0;  Diff_BBestGen=0;
       PowDemand=PowerDemand[Hour];
       write_xy(1,1,"");printf("You're running GA/SA %s",InpFileName);
        /* Watch.Start; */
       Start_Time();

       Species[0].NMicro = 0;
       Species[0].NCross = 0;
       Species[0].NMutation = 0;
       Species[0].BestFit.Generation=0;
       Species[0].BestFit.Fitness=0.0;

       BBest.Generation=0;  BBest.Fitness=0.0;  BBest.Who=0;
         /* Compute Ramp Rate Constraint */
       SumPr[min]=0;        SumPr[max]=0;
       for (i=0; i < NUnit; i++)
       {
               Pr[i][min]= P[i]-Dr[i];  Pr[i][max]  = P[i]+Ur[i];
               /* If it is out of limit */
               if (Pr[i][min]< Pi[i][min])  Pr[i][min]  = Pi[i][min];
               if (Pr[i][max] >Pi[i][max])  Pr[i][max]  = Pi[i][max];
                 /* Check sum of power (Pr) in constraint */
                 SumPr[min] += Pr[i][min];    SumPr[max] += Pr[i][max];
         }

       srand(mseed);
       SumPow  = Initial(SumPr[max],PowDemand,P);
       Cost = SA(PowDemand,P,1000,100,1,&SumPow,&P_Loss,&ref);

        /*-----------SGA----------------*/
        warmup_random(SeedRandom);
        InitPop(&Species[0],SumPow,P);
        do{    Generation++;
               SGA_Process(RoulletWhl,PopSize,RateConv,&Species[0] );
        } while (Generation < MaxGeneration);

        Stop_Time();
        Report(Hour);

        TotalCost += Cost;

  } /*------- end of Step Hour loop --------*/
  TotalCost /= 4;
```

```
                fprintf(outfp,"SumCPUTime %li seconds %li
microseconds\n",SumTime.tv_sec,SumTime.tv_usec);

                if ((TotalCost<BestTCost) || (mseed==First_mseed))
                { BestTCost=TotalCost;          BestFile=mseed;}
                 fprintf(outfp,"Total Oper Cost =%20.4f\n", TotalCost);
                 fprintf(outfp,"Best FileNumber =%i\n",      BestFile);
                 fprintf(outfp,"Best Total Cost =%20.4f\n", BestTCost);
                 fflush(outfp);
                fclose(outfp);

              /*------ Summary ------*/
              fprintf(Summary,"\nSeed no.%2d\t %.4lf",mseed,TotalCost);
               SummaryCost += TotalCost;

        } /*------- end of mseed loop --------*/

        fprintf(Summary,"\nAvg Cost %.4lf",(SummaryCost/Last_mseed));
        fprintf(Summary,"\nBestCost %.4lf",BestTCost);
        fclose(Summary);
        freeall_sga();
        freeall_sa();
        system(cls);
        return 0;

}
```