Daniel Lulka Vargas 00190585 Algoritmos de Optimización 4/05/2017 Maestro Miguel Ángel Mendez Mendez Tarea 3

Algoritmos Genéticos

OBJETIVO:

Realizar el problema de Giappetto mediante algoritmos genéticos.

PLANTEAMIENTO:

1. Tamaño de los cromosomas de cada individuo
2. Tamaño de la población
3. Obtener en cuantas generaciones se obtuvo el resultado

DESARROLLO:

CÓDIGO:

#include <iostream> //cin, cout

#include <stdlib.h> //srand, rand

#include <time.h> //srand, rand time

using namespace std;

typedef unsigned char tipo;

char juguetes[][20]= {"trenes","soldados"};

int getbit(tipo x, int pos){

 return(x & (1<<pos) ?1:0);

}

void setbit(tipo & x, int pos, int val){

 x= val? x|(1<<pos): x & ~(1<<pos);

}

void  showbits(tipo x){

   int i;

   cout<<(int)x<<" en binario es:\n";

   for (i=0; i<8; i++)

     if (getbit(x,i)) cout<<"1";

     else cout<<"0";

   cout<<"\n";

}

int fitness(tipo \* cromosoma){

int value;

if((cromosoma[0]+cromosoma[1])>80){

value = 0;

}else if((cromosoma[0]+ 2\*cromosoma[1])>100){

value=0;

}else if(cromosoma[1]>40 || cromosoma[0]>80){

value=0;

}else{

value = 2\*cromosoma[0] + 3\*cromosoma[1];

}

return value;

}

void muestraindividuo(tipo \*cromosoma){// nos sirve para visualizar los 7 cromosomas del individuo

int j;//contador de cromosomas

for(j=0; j<2;j++){// nos despliega los genes 0 1 0 ...

cout<<j<<":"<<(int)cromosoma[j]<<" ";

}

cout<<endl;

//cout<<juguetes[j];

//cout<<"\t";

//cout<<endl;

}

main(){

int i,j,g,pos;// individuos, cromosomas, generaciones(contadores)

int poblacion=500;//numero de individuos por generación

       int ncromosoma=2;

tipo cromosoma[poblacion][ncromosoma];//individuos con 2 cromosomas cada uno

int best=0;

srand(time(NULL));//inicializamos generador de numeros aleatorios

for(i=0;i<poblacion;i++){//inicializamos la poblacion

for(j=0;j<ncromosoma; j++){

cromosoma[i][j]=rand()%100; // generamos la población inicial de 20

cout<<(int)cromosoma[i][j];

cout<<" ";

}

cout<<endl;

}

for(g=0;g<100;g++){//100 generaciones maximo

cout<<"Generacion:"<<g<<endl;

for(i=0;i<poblacion;i++){

if (fitness(cromosoma[best])<fitness(cromosoma[i]))

best=i;

}

for(i=0;i<poblacion;i++){

if(i!=best){//si es diferente del mejor

for(j=0;j<2;j++){ //CRUZAMIENTO

if(rand()%100<50)

cromosoma[i][j]=cromosoma[best][j];

else

cromosoma[i][j]=cromosoma[i][j];

if(rand()%100<4){

pos=rand()%8;

setbit(cromosoma[i][j], pos, rand()%2);

}

}//for cromosomas

}//if

}//for poblacion

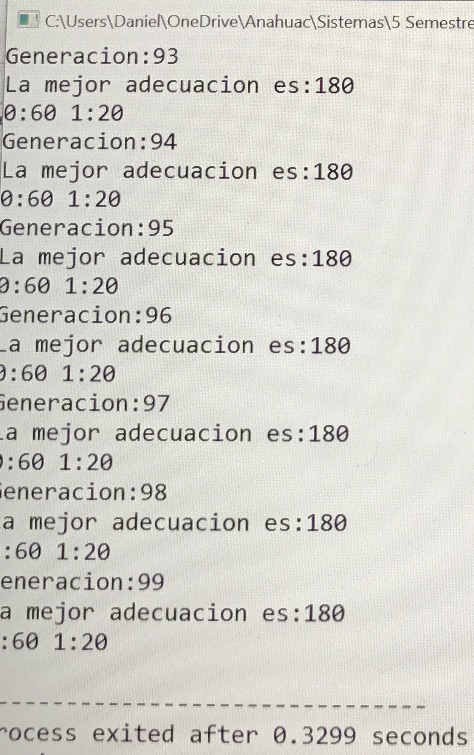
cout<<"La mejor adecuacion es:"<<fitness(cromosoma[best])<<endl;

muestraindividuo(cromosoma[best]);//muestra individuo

}//fin evolucion

}//main

EJECUCIÓN:



Conclusión:

Gracias al algoritmo genético pudimos resolver el problema de Giappetto de una manera más rápida y eficiente, ya que, a pesar de que al utilizarlo dependemos del azar, con cada generación de va seleccionando a la mejor combinación y como se vuelve a cruzar, siempre vamos quedándonos con el mejor de cada generación, solo tenemos que asegurarnos que la población sea lo suficientemente grande.