



INSTITUTO POLITÉCNICO NACIONAL

**ESCUELA SUPERIOR DE
CÓMPUTO**



GENETIC ALGORITHMS

REPORTE PRACTICA IV

“SELECCIÓN POR TORNEO”

ALUMNO: SOLIS SANCHEZ JHOVANY

GRUPO: 3CM5

BOLETA: 2015630489

INTRODUCCIÓN

La selección mediante torneo es similar a la de jerarquías en términos de la presión de selección. La idea básica del método es seleccionar con base en comparaciones directas de los individuos.

Hay 2 versiones de la selección mediante torneo:

- Determinista.
- Probabilística.

El algoritmo de la versión determinista es el siguiente:

- Barajar los individuos de la población.
- Escoger un número p de individuos (típicamente 2).
- Compararlos con base en su aptitud.
- El ganador del “torneo” es el individuo más apto.
- Debe barajarse la población un total de p veces para seleccionar N padres (donde N es el tamaño de la población).

El algoritmo de la versión probabilística es idéntico al anterior, excepto por el paso en que se escoge al ganador. En vez de seleccionar siempre al individuo con aptitud más alta, se aplica $\text{flip}(p)$ y si el resultado es cierto, se selecciona al más apto. De lo contrario, se selecciona al menos apto. El valor de p permanece fijo a lo largo de todo el proceso evolutivo y se escoge comúnmente en el siguiente rango $0.5 < p \leq 1$, el cual para este ejercicio usaremos $0.7 \geq p \leq 1$

DESARROLLO

Con el fin de demostrar el funcionamiento del programa se utilizara una población de 16 individuos variando la cantidad de generaciones de en 5, 30,50 y 100 generaciones que se obtendrán. Al iniciar la ejecución este generara una serie de individuos compuestos por 4 bits y obtendrá el su valor en decimal.

Individuos	x	fitness	Probabilidad	P. Acumulada
1 0 1 0	10	3	0.0938	0.0938
0 0 1 0	2	1	0.0312	0.1250
1 1 1 1	15	3	0.0938	0.2188
0 1 1 0	6	0	0.0000	0.2188
1 0 1 1	11	5	0.1562	0.3750
0 1 1 0	6	0	0.0000	0.3750
0 1 1 0	6	0	0.0000	0.3750
1 1 1 0	14	3	0.0938	0.4688
1 1 1 0	14	3	0.0938	0.5625
0 0 1 0	2	1	0.0312	0.5938
1 0 1 0	10	3	0.0938	0.6875
1 0 1 1	11	5	0.1562	0.8438
1 1 1 0	14	3	0.0938	0.9375
0 0 1 0	2	1	0.0312	0.9688
0 0 1 0	2	1	0.0312	1.0000
0 1 1 0	6	0	0.0000	1.0000

Illustration 1: Generación 0

Mostrara la el resultado de aplicarle la función de fitness la cual para este algoritmo sera:

$$f(x) = \text{ABS} \left| \frac{x - 5}{2 + \text{Sen}(x)} \right|$$

el cuadrado del valor decimal de cada individuo, mostrara la probabilidad o valor esperado a si como la probabilidad acumulada.

```
Suma : 173619
Promedio : 21702.3750
Probabilidad Maxima : 0.2589
Probabilidad Minima : 0.0092
```

Illustration 2: Generación 0 - datos

Realizara el calculo correspondiente a la suma total de los resultados obtenidos de la función fitness, el promedio de estos, así como la probabilidad máxima y mínima de su generación.

Continuando con el algoritmo se generaran números coma realizara un barajeo el cual ordenara en un arreglo los individuos que se enfrentaran y se seleccionaran a los ganadores a través de la función flip.

```
void barajear(int bar[]){
    /* Barajeo */
    int rI=0,tB=0, fg=0;
    for(int b=0;b<TAM;b++){
        rI = randomValueIndex(TAM);
        if(rI==0) rI=1;
        for(int c=0;c<tB+1;c++){
            if(rI==bar[c]){ fg=0; break; }
            else fg=1;
        }
        if(fg==1) { bar[tB]=rI; fg=0; tB++; }
        else if(b<TAM-1) b--;
    }
}
```

Illustration 3: barajeo

```
/* Seleccion de posibles candidatos */
int t=0, w=0, l=0;
for(int i=0;i<2;i++){
    barajear(bar);
    for(int j=0;j<TAM/2;j++){
        rand = randomValue();
        if(rand<0.4) w=0;
        else w=1;
        int index = bar[l+w];
        memcpy(aux[t++],poblacion[index],bits);
        l+=2;
    }
    l=0;
}
```

Illustration 4: seleccion

Después se seguirá aplicando el algoritmo de la ruleta para las siguientes generaciones y la única varianza en la forma en que son selección los individuos.

5 Generaciones.

Algoritmo Genetico Simple con 4 Generaciones

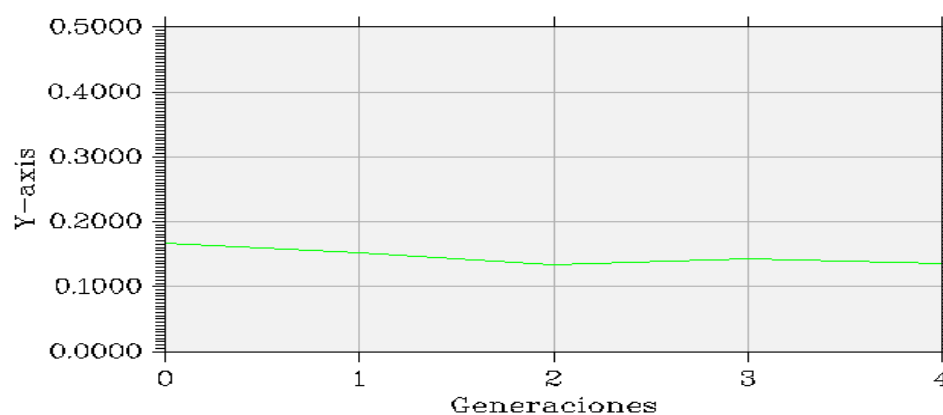


Illustration 5: 5 Generaciones

10 Generaciones.

Algoritmo Genetico Simple con 9 Generaciones

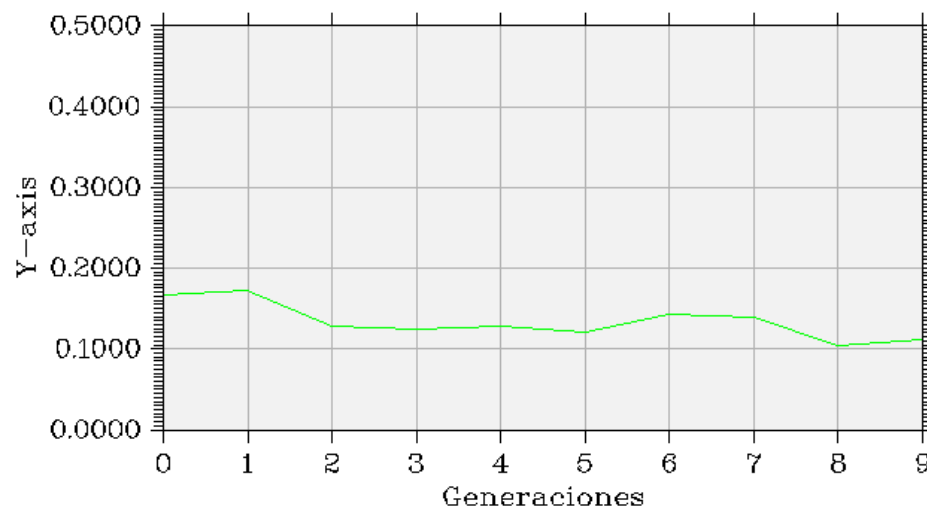


Illustration 6: 10 Generaciones

30 Generaciones.

Algoritmo Genetico Simple con 29 Generaciones

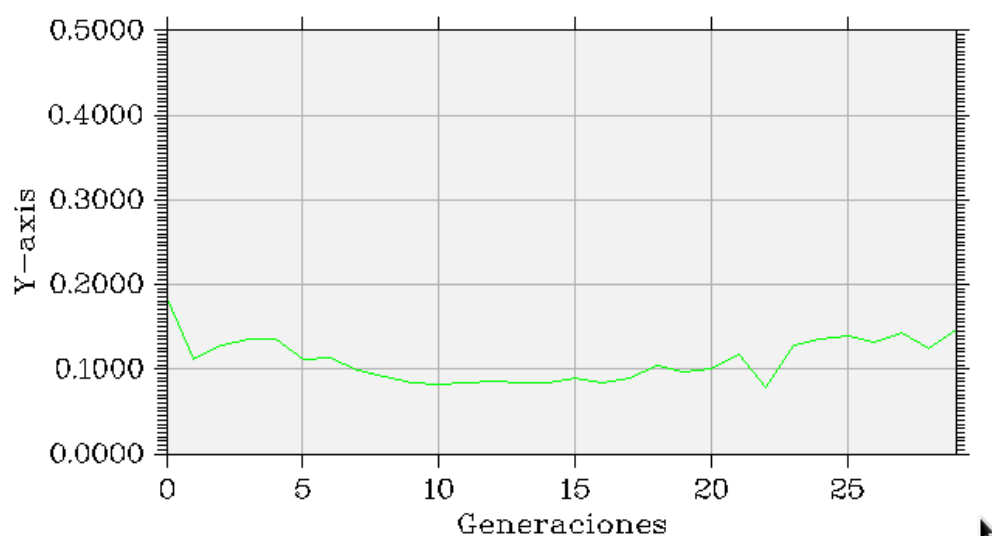


Illustration 7: 30 Generaciones

50 Generaciones

Algoritmo Genetico Simple con 49 Generaciones

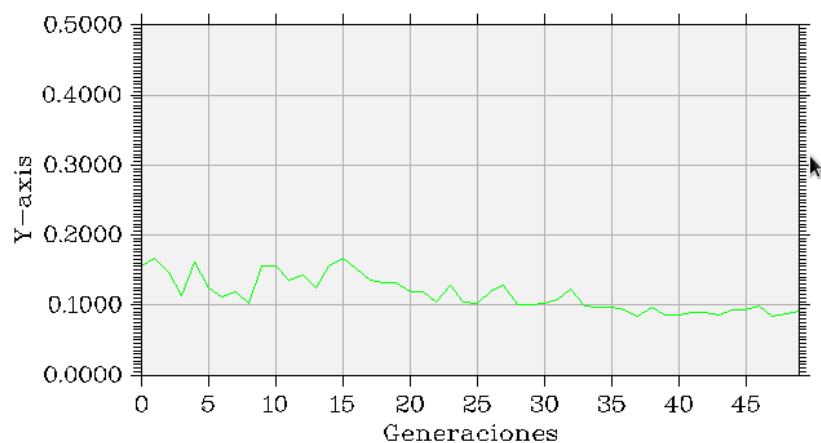


Illustration 8: 50 Generaciones

100 Generaciones

Algoritmo Genetico Simple con 99 Generaciones

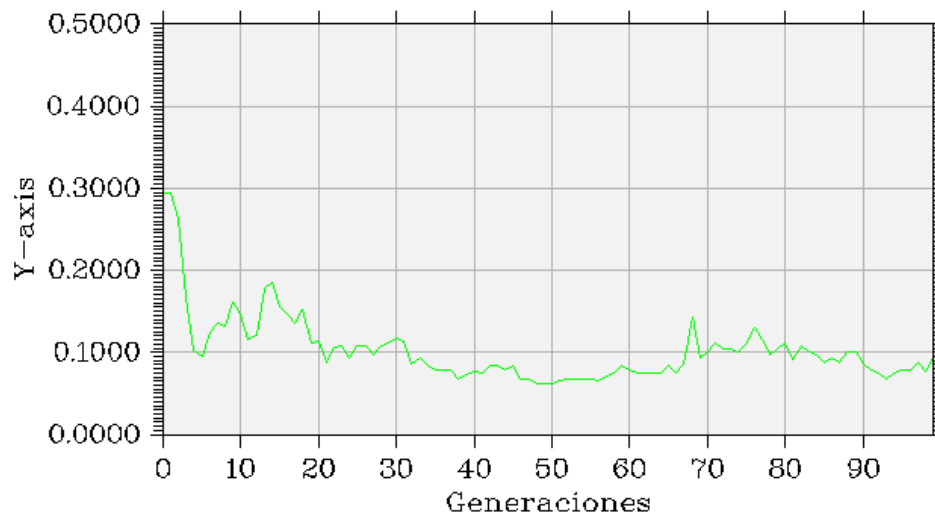


Illustration 9: 100 Generaciones

Como se puede observar tanto la gráfica el promedio máximo en color verde como el promedio mínimo en color rojo tienden a converger ambas gráficas conforme mas generaciones pasasen. Cuanto mas grande es la cantidad de generaciones a tomar en cuenta el algoritmo empieza a realizar cambios extraños en los individuos.

CONCLUSIÓN

Como se puede ver en algunos alguna gráficas se muestran algunos cambios en su trayectoria pero al final seguirán convergiendo mientras mas generaciones sean tomadas en cuenta, en algunos casos cuando la población aumentada al igual que las generación la maquina tardaba para realizar los cálculos y algunas veces mostraba algunas picos en las gráficas o cambiaba completamente la trayectoria de esta. A mi parecer la mutación provocaba esos cambios. El algoritmo tendré que mejorar para resolver algunos conflictos de memoria que se provocan en alguna ocasiones.