

Universidad Autónoma Metropolitana  
Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación  
Proyecto de Investigación

Análisis comparativo de técnicas de cruce adaptativa en algoritmos  
genéticos

Peñaloza Plata Gonzalo  
210200230

Trimestre 2014 Invierno  
25 de marzo de 2014

Dr. Antonin Sébastien Ponsich  
Profesor Asociado  
Departamento de Sistemas

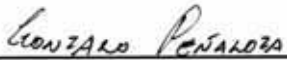
Yo, Dr. Antonin Sébastien Ponsich, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Firma del asesor

Yo, Gonzalo Peñaloza Plata, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Firma del alumno

Cuando estás solucionando un problema, no te preocupes. Ahora, después de que has resuelto el problema es el momento de preocuparse.

**Richard Feynman**

Dicen que triste cosa es no tener amigos,  
Pero más triste es no tener enemigos.  
Porque quien enemigos no tenga,  
Es señal de que no tiene,  
Ni talento que haga sombra,  
Ni bienes que se le codicien,  
Ni carácter que impresione,  
Ni valor temido,  
Ni honra de la que se murmure,  
Ni ninguna cosa buena que se le envidie.

**José Martí**

## Resumen

Los algoritmos genéticos son una técnica metaheurística inspirada en los conceptos de la teoría de la evolución. Imitando los procesos de la evolución de las especies, estos algoritmos son capaces de crear soluciones para problemas del mundo real. El desempeño de un algoritmo genético depende de la definición y aplicación de operadores genéticos, tales como la cruce y la mutación.

El operador de cruce es el encargado de combinar el material genético de los miembros presentes en una población con el fin de generar nuevos individuos. Desde la aparición de los algoritmos genéticos en la segunda mitad del siglo pasado, hasta la fecha, se han propuesto diferentes técnicas de cruce. Técnicas como la cruce de un punto, la cruce de dos puntos y la cruce discreta, fueron de las primeras técnicas en aparecer, por lo que su uso y conocimiento se encuentran bastante difundidos.

En años más recientes diferentes investigadores han propuesto nuevas técnicas de cruce, conocidas como adaptativas. Estas técnicas aprovechan las características de los miembros de la población para establecer la forma en que se combina el material genético de los padres. A través del paso de las generaciones, estas técnicas revisan el estado de la población y actualizan el valor de sus parámetros de operación con el objetivo de obtener individuos que aprovechen de mejor manera las características de sus padres y crear individuos más aptos.

En este trabajo se presentan algunas técnicas de cruce adaptativa publicadas en años recientes y se utilizan para la búsqueda de la solución de un conjunto de funciones de prueba ampliamente conocidas. El objetivo es comparar los resultados obtenidos al implementar las técnicas seleccionadas dentro de un algoritmo genético simple y analizar como se ve afectado el tiempo de ejecución y la calidad de solución arrojada por el algoritmo.

De acuerdo a los experimentos numéricos realizados, resulta difícil decantarse hacia alguna técnica en particular, pues los valores presentados para la función objetivo son sumamente similares así como los tiempos de ejecución. Caso aparte el de la técnica XHC, la cual presenta excelentes resultados, a cambio de un aumento significativo en el tiempo de ejecución.

# Índice

<b>Resumen</b>	<b>II</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de tablas</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Justificación</b>	<b>2</b>
<b>3. Antecedentes</b>	<b>3</b>
3.1. Historia de la computación evolutiva . . . . .	5
<b>4. Objetivos</b>	<b>7</b>
4.1. General . . . . .	7
4.2. Específicos . . . . .	7
<b>5. Marco teórico</b>	<b>8</b>
5.1. Definición . . . . .	8
5.2. Estructura básica . . . . .	8
5.3. Parámetros en un algoritmo genético . . . . .	9
5.4. Codificación de individuos . . . . .	10
5.5. Selección . . . . .	12
5.6. Operadores genéticos . . . . .	12
5.7. Ventajas y desventajas . . . . .	15
<b>6. Desarrollo del proyecto</b>	<b>16</b>
6.1. Implementación . . . . .	18
<b>7. Resultados</b>	<b>21</b>
7.1. Experimentos numéricos . . . . .	21
7.1.1. Codificación binaria . . . . .	21
7.1.2. Codificación real . . . . .	23
7.2. Tiempo de ejecución . . . . .	25
7.2.1. Codificación binaria . . . . .	25
7.2.2. Codificación real . . . . .	28
<b>8. Análisis y discusión de resultados</b>	<b>32</b>
<b>9. Conclusiones</b>	<b>33</b>
<b>Referencias</b>	<b>34</b>

<b>Apéndices</b>	<b>37</b>
<b>A. Funciones de prueba</b>	<b>37</b>
<b>B. Tablas de resultados</b>	<b>43</b>
B.1. Representación binaria . . . . .	43
B.1.1. Ackley & Cruza de 2 puntos . . . . .	43
B.1.2. Ackley & PAX . . . . .	45
B.1.3. Ackley & SANUX . . . . .	47
B.1.4. Ackley & Selective Crossover . . . . .	49
B.1.5. Esfera & Cruza de 2 puntos . . . . .	51
B.1.6. Esfera & PAX . . . . .	53
B.1.7. Esfera & SANUX . . . . .	55
B.1.8. Esfera & Selective Crossover . . . . .	57
B.1.9. Griewank & Cruza de 2 puntos . . . . .	59
B.1.10. Griewank & PAX . . . . .	61
B.1.11. Griewank & SANUX . . . . .	63
B.1.12. Griewank & Selective Crossover . . . . .	65
B.1.13. Rastrigin & Cruza de 2 puntos . . . . .	67
B.1.14. Rastrigin & PAX . . . . .	69
B.1.15. Rastrigin & SANUX . . . . .	71
B.1.16. Rastrigin & Selective Crossover . . . . .	73
B.1.17. Rosenbrock & Cruza de 2 puntos . . . . .	75
B.1.18. Rosenbrock & PAX . . . . .	77
B.1.19. Rosenbrock & SANUX . . . . .	79
B.1.20. Rosenbrock & Selective Crossover . . . . .	81
B.1.21. Schwefel & Cruza de 2 puntos . . . . .	83
B.1.22. Schwefel & PAX . . . . .	85
B.1.23. Schwefel & SANUX . . . . .	87
B.1.24. Schwefel & Selective Crossover . . . . .	89
B.2. Representación real . . . . .	91
B.2.1. Ackley & GPAX . . . . .	91
B.2.2. Ackley & Recombinación intermedia . . . . .	93
B.2.3. Ackley & SBX . . . . .	95
B.2.4. Ackley & SBX adaptativo . . . . .	97
B.2.5. Ackley & XHC . . . . .	99
B.2.6. Esfera & GPAX . . . . .	101
B.2.7. Esfera & Recombinación intermedia . . . . .	103
B.2.8. Esfera & SBX . . . . .	105
B.2.9. Esfera & SBX adaptativo . . . . .	107
B.2.10. Esfera & XHC . . . . .	109
B.2.11. Griewank & GPAX . . . . .	111

B.2.12. Griewank & Recombinación intermedia . . . . .	113
B.2.13. Griewank & SBX . . . . .	115
B.2.14. Griewank & SBX adaptativo . . . . .	117
B.2.15. Griewank & XHC . . . . .	119
B.2.16. Rastringin & GPAX . . . . .	121
B.2.17. Rastringin & Recombinación intermedia . . . . .	123
B.2.18. Rastringin & SBX . . . . .	125
B.2.19. Rastringin & SBX Adaptativo . . . . .	127
B.2.20. Rastringin & XHC . . . . .	129
B.2.21. Rosenbrock & GPAX . . . . .	131
B.2.22. Rosenbrock & Recombinación intermedia . . . . .	133
B.2.23. Rosenbrock & SBX . . . . .	135
B.2.24. Rosenbrock & SBX Adaptativo . . . . .	137
B.2.25. Rosenbrock & XHC . . . . .	139
B.2.26. Schwefel & GPAX . . . . .	141
B.2.27. Schwefel & Recombinación intermedia . . . . .	143
B.2.28. Schwefel & SBX . . . . .	145
B.2.29. Schwefel & SBX Adaptativo . . . . .	147
B.2.30. Schwefel & XHC . . . . .	149
<b>C. Código fuente</b>	<b>151</b>
C.1. Algoritmo genético . . . . .	151
C.1.1. Generación de la población inicial . . . . .	151
C.1.2. Selección de los supervivientes . . . . .	152
C.1.3. Mutación . . . . .	152
C.1.4. Elitismo . . . . .	153
C.2. Funciones de prueba . . . . .	154
C.2.1. Ackley . . . . .	154
C.2.2. Esfera . . . . .	154
C.2.3. Griewank . . . . .	154
C.2.4. Rastringin . . . . .	154
C.2.5. Rosenbrock . . . . .	155
C.2.6. Schwefel . . . . .	155
C.3. Técnicas de cruza . . . . .	156
C.3.1. Cruza de 2 puntos . . . . .	156
C.3.2. PAX . . . . .	157
C.3.3. SANUX . . . . .	159
C.3.4. Selective Crossover . . . . .	161
C.3.5. GPAX . . . . .	163
C.3.6. Recombinación intermedia . . . . .	167
C.3.7. SBX . . . . .	167
C.3.8. SBX Adaptativo . . . . .	168

C.3.9. XHC . . . . .	170
C.4. Ejemplo completo . . . . .	172
C.4.1. Función de Rastringin y PAX . . . . .	172
C.4.2. Función de Griewank y SBX Adaptativo . . . . .	181



## Lista de figuras

1.	Caricatura de Darwin publicada por la revista Hornet en 1871. . . . .	4
2.	Estructura básica de un algoritmo genético simple. . . . .	9
3.	Ejemplo de cruce de un punto. . . . .	13
4.	Ejemplo de cruce de dos puntos. . . . .	13
5.	Descripción básica de los módulos del proyecto. . . . .	16
6.	Tiempo de ejecución para la función de Ackley con codificación binaria. . . . .	25
7.	Tiempo de ejecución para la función Esfera con codificación binaria. . . . .	26
8.	Tiempo de ejecución para la función de Griewank con codificación binaria. . . . .	26
9.	Tiempo de ejecución para la función de Rastrigin con codificación binaria. . . . .	27
10.	Tiempo de ejecución para la función de Rosenbrock con codificación binaria. . . . .	27
11.	Tiempo de ejecución para la función de Schwefel con codificación binaria. . . . .	28
12.	Tiempo de ejecución para la función de Ackley con codificación real. . . . .	28
13.	Tiempo de ejecución para la función Esfera con codificación real. . . . .	29
14.	Tiempo de ejecución para la función de Griewank con codificación real. . . . .	29
15.	Tiempo de ejecución para la función de Rastrigin con codificación real. . . . .	30
16.	Tiempo de ejecución para la función de Rosenbrock con codificación real. . . . .	30
17.	Tiempo de ejecución para la función de Schwefel con codificación real. . . . .	31
18.	Gráfica de la función de Rosenbrock para dos variables. . . . .	37
19.	Gráfica de la función de Rastrigin para dos variables. . . . .	38
20.	Gráfica de la función esfera para dos variables. . . . .	39
21.	Gráfica de la función de Schwefel para dos variables. . . . .	40
22.	Gráfica de la función de Ackley para dos variables. . . . .	41
23.	Gráfica de la función de Griewank para dos variables. . . . .	42

## Lista de tablas

1.	Resultados para la función de Ackley con codificación binaria. . . . .	21
2.	Resultados para la función Esfera con codificación binaria. . . . .	21
3.	Resultados para la función de Griewank con codificación binaria. . . . .	21
4.	Resultados para la función de Rastrigin con codificación binaria. . . . .	22
5.	Resultados para la función de Rosenbrock con codificación binaria. . . . .	22
6.	Resultados para la función de Schwefel con codificación binaria. . . . .	22
7.	Resultados para la función de Ackley con codificación real. . . . .	23
8.	Resultados para la función Esfera con codificación real. . . . .	23
9.	Resultados para la función de Griewank con codificación real. . . . .	23
10.	Resultados para la función de Rastrigin con codificación real. . . . .	24
11.	Resultados para la función de Rosenbrock con codificación real. . . . .	24
12.	Resultados para la función de Schwefel con codificación real. . . . .	24
13.	Resultados para la función de Ackley utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	43
14.	Resultados para la función de Ackley utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	44
15.	Resultados para la función de Ackley utilizando PAX y una semilla de aleatorios fija. . . . .	45
16.	Resultados para la función de Ackley utilizando PAX y una semilla de aleatorios variable. . . . .	46
17.	Resultados para la función de Ackley utilizando SANUX y una semilla de aleatorios fija. . . . .	47
18.	Resultados para la función de Ackley utilizando SANUX y una semilla de aleatorios variable. . . . .	48
19.	Resultados para la función de Ackley utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	49
20.	Resultados para la función de Ackley utilizando Selective Crossover y una semilla de aleatorios variable. . . . .	50
21.	Resultados para la función de Esfera utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	51
22.	Resultados para la función Esfera utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	52
23.	Resultados para la función de Esfera utilizando PAX y una semilla de aleatorios fija. . . . .	53
24.	Resultados para la función Esfera utilizando PAX y una semilla de aleatorios variable. . . . .	54
25.	Resultados para la función de Esfera utilizando SANUX y una semilla de aleatorios fija. . . . .	55
26.	Resultados para la función Esfera utilizando PAX y una semilla de aleatorios variable. . . . .	56
27.	Resultados para la función de Esfera utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	57
28.	Resultados para la función Esfera utilizando Selective Crossover y una semilla de aleatorios variable. . . . .	58

29.	Resultados para la función de Griewank utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	59
30.	Resultados para la función de Griewank utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	60
31.	Resultados para la función de Griewank utilizando PAX y una semilla de aleatorios fija. . . . .	61
32.	Resultados para la función de Griewank utilizando PAX y una semilla de aleatorios variable. . . . .	62
33.	Resultados para la función de Griewank utilizando SANUX y una semilla de aleatorios fija. . . . .	63
34.	Resultados para la función de Griewank utilizando SANUX y una semilla de aleatorios variable. . . . .	64
35.	Resultados para la función de Griewank utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	65
36.	Resultados para la función de Griewank utilizando Selective Crossover y una semilla de aleatorios variable. . . . .	66
37.	Resultados para la función de Rastringin utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	67
38.	Resultados para la función de Rastringin utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	68
39.	Resultados para la función de Rastringin utilizando PAX y una semilla de aleatorios fija. . . . .	69
40.	Resultados para la función de Rastringin utilizando PAX y una semilla de aleatorios variable. . . . .	70
41.	Resultados para la función de Rastringin utilizando SANUX y una semilla de aleatorios fija. . . . .	71
42.	Resultados para la función de Rastringin utilizando SANUX y una semilla de aleatorios variable. . . . .	72
43.	Resultados para la función de Rastringin utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	73
44.	Resultados para la función de Rastringin utilizando Selective Crossover y una semilla de aleatorios variable. . . . .	74
45.	Resultados para la función de Rosenbrock utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	75
46.	Resultados para la función de Rosenbrock utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	76
47.	Resultados para la función de Rosenbrock utilizando PAX y una semilla de aleatorios fija. . . . .	77
48.	Resultados para la función de Rosenbrock utilizando PAX y una semilla de aleatorios variable. . . . .	78
49.	Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios fija. . . . .	79

50.	Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios variable. . . . .	80
51.	Resultados para la función de Rosenbrock utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	81
52.	Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios variable. . . . .	82
53.	Resultados para la función de Schwefel utilizando Cruza de 2 puntos y una semilla de aleatorios fija. . . . .	83
54.	Resultados para la función de Schwefel utilizando Cruza de 2 puntos y una semilla de aleatorios variable. . . . .	84
55.	Resultados para la función de Schwefel utilizando PAX y una semilla de aleatorios fija. . . . .	85
56.	Resultados para la función de Schwefel utilizando PAX y una semilla de aleatorios variable. . . . .	86
57.	Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios fija. . . . .	87
58.	Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios variable. . . . .	88
59.	Resultados para la función de Schwefel utilizando Selective Crossover y una semilla de aleatorios fija. . . . .	89
60.	Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios variable. . . . .	90
61.	Resultados para la función de Ackley utilizando GPAX y una semilla de aleatorios fija. . . . .	91
62.	Resultados para la función de Ackley utilizando GPAX y una semilla de aleatorios variable. . . . .	92
63.	Resultados para la función de Ackley utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	93
64.	Resultados para la función de Ackley utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	94
65.	Resultados para la función de Ackley utilizando SBX y una semilla de aleatorios fija. . . . .	95
66.	Resultados para la función de Ackley utilizando SBX y una semilla de aleatorios variable. . . . .	96
67.	Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios fija. . . . .	97
68.	Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios variable. . . . .	98
69.	Resultados para la función de Ackley utilizando XHC y una semilla de aleatorios fija. . . . .	99
70.	Resultados para la función de Ackley utilizando XHC y una semilla de aleatorios variable. . . . .	100
71.	Resultados para la función Esfera utilizando GPAX y una semilla de aleatorios fija. . . . .	101

72.	Resultados para la función Esfera utilizando GPAX y una semilla de aleatorios variable. . . . .	102
73.	Resultados para la función Esfera utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	103
74.	Resultados para la función Esfera utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	104
75.	Resultados para la función Esfera utilizando SBX y una semilla de aleatorios fija. .	105
76.	Resultados para la función Esfera utilizando SBX y una semilla de aleatorios variable.	106
77.	Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios fija. . . . .	107
78.	Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios variable. . . . .	108
79.	Resultados para la función Esfera utilizando XHC y una semilla de aleatorios fija. .	109
80.	Resultados para la función Esfera utilizando XHC y una semilla de aleatorios variable.	110
81.	Resultados para la función de Griewank utilizando GPAX y una semilla de aleatorios fija. . . . .	111
82.	Resultados para la función de Griewank utilizando GPAX y una semilla de aleatorios variable. . . . .	112
83.	Resultados para la función de Griewank utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	113
84.	Resultados para la función de Griewank utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	114
85.	Resultados para la función de Griewank utilizando SBX y una semilla de aleatorios fija. . . . .	115
86.	Resultados para la función de Griewank utilizando SBX y una semilla de aleatorios variable. . . . .	116
87.	Resultados para la función de Griewank utilizando SBX adaptativo y una semilla de aleatorios fija. . . . .	117
88.	Resultados para la función de Griewank utilizando SBX adaptativo y una semilla de aleatorios variable. . . . .	118
89.	Resultados para la función de Griewank utilizando XHC y una semilla de aleatorios fija. . . . .	119
90.	Resultados para la función de Griewank utilizando XHC y una semilla de aleatorios variable. . . . .	120
91.	Resultados para la función de Rastrigin utilizando GPAX y una semilla de aleatorios fija. . . . .	121
92.	Resultados para la función de Rastrigin utilizando GPAX y una semilla de aleatorios variable. . . . .	122
93.	Resultados para la función de Rastrigin utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	123
94.	Resultados para la función de Rastrigin utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	124

95.	Resultados para la función de Rastringin utilizando SBX y una semilla de aleatorios fija. . . . .	125
96.	Resultados para la función de Rastringin utilizando SBX y una semilla de aleatorios variable. . . . .	126
97.	Resultados para la función de Rastringin utilizando SBX Adaptativo y una semilla de aleatorios fija. . . . .	127
98.	Resultados para la función de Rastringin utilizando SBX Adaptativo y una semilla de aleatorios variable. . . . .	128
99.	Resultados para la función de Rastringin utilizando XHC y una semilla de aleatorios fija. . . . .	129
100.	Resultados para la función de Rastringin utilizando XHC y una semilla de aleatorios variable. . . . .	130
101.	Resultados para la función de Rosenbrock utilizando GPAX y una semilla de aleatorios fija. . . . .	131
102.	Resultados para la función de Rosenbrock utilizando GPAX y una semilla de aleatorios variable. . . . .	132
103.	Resultados para la función de Rosenbrock utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	133
104.	Resultados para la función de Rosenbrock utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	134
105.	Resultados para la función de Rosenbrock utilizando SBX y una semilla de aleatorios fija. . . . .	135
106.	Resultados para la función de Rosenbrock utilizando SBX y una semilla de aleatorios variable. . . . .	136
107.	Resultados para la función de Rosenbrock utilizando SBX Adaptativo y una semilla de aleatorios fija. . . . .	137
108.	Resultados para la función de Rosenbrock utilizando SBX Adaptativo y una semilla de aleatorios variable. . . . .	138
109.	Resultados para la función de Rosenbrock utilizando XHC y una semilla de aleatorios fija. . . . .	139
110.	Resultados para la función de Rosenbrock utilizando XHC y una semilla de aleatorios variable. . . . .	140
111.	Resultados para la función de Schwefel utilizando GPAX y una semilla de aleatorios fija. . . . .	141
112.	Resultados para la función de Schwefel utilizando GPAX y una semilla de aleatorios variable. . . . .	142
113.	Resultados para la función de Schwefel utilizando Recombinación intermedia y una semilla de aleatorios fija. . . . .	143
114.	Resultados para la función de Schwefel utilizando Recombinación intermedia y una semilla de aleatorios variable. . . . .	144
115.	Resultados para la función de Schwefel utilizando SBX y una semilla de aleatorios fija. . . . .	145

116. Resultados para la función de Schwefel utilizando SBX y una semilla de aleatorios variable. . . . .	146
117. Resultados para la función de Schwefel utilizando SBX Adaptativo y una semilla de aleatorios fija. . . . .	147
118. Resultados para la función de Schwefel utilizando SBX Adaptativo y una semilla de aleatorios variable. . . . .	148
119. Resultados para la función de Schwefel utilizando XHC y una semilla de aleatorios fija. . . . .	149
120. Resultados para la función de Schwefel utilizando XHC y una semilla de aleatorios variable. . . . .	150

## 1. Introducción

La complejidad que caracteriza un problema es un factor determinante para escoger la técnica de búsqueda de la solución. Cuando se trata de resolver problemas de optimización que pertenecen a la clase NP-difícil<sup>1</sup>, el costo computacional requerido para encontrar la solución óptima puede ser muy alto y no se puede siempre garantizar la optimalidad global de la solución encontrada (problemas continuos no-lineales). Se pueden mencionar como ejemplos, algunos problemas combinatorios NP-difíciles (problema del agente viajero o de la mochila) o algunas funciones de prueba continuas incluyendo una gran cantidad de óptimos locales (por ejemplo, función de Rastringin). Una alternativa común, en estos casos, consiste en aplicar métodos metaheurísticos, para encontrar una solución aproximada con un costo computacional razonable.

Los algoritmos genéticos son una técnica metaheurística que puede usarse para resolver problemas de búsqueda y optimización cuya complejidad es alta. Su modo operativo está basado en una analogía con los conceptos del neodarwinismo (esencialmente basado en la teoría de la evolución propuesta por Darwin en 1859). A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza, vista como un ambiente con recursos limitados, de acuerdo a los principios de la selección natural, la supervivencia de los más adaptados, la transmisión y modificación del patrimonio genético. Imitando estos procesos, los algoritmos genéticos son capaces de ir creando y evolucionando soluciones de buena calidad para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas, así como de operadores de variación apropiados.

Un operador sumamente importante en el funcionamiento de un algoritmo genético es la *cruza*, la cual permite la obtención de nuevos individuos a partir de los ya existentes. Estos nuevos individuos se distinguen por compartir características con sus progenitores, por lo que la selección de una técnica de cruce adecuada permite que los nuevos individuos de la población hereden las mejores características y converjan a una solución adecuada en una menor cantidad de generaciones. El funcionamiento de este operador suele ser determinado por parámetros fijos difíciles de ajustar, por lo que una estrategia autoadaptativa presenta el interés de que la técnica de búsqueda aprende por sí misma cómo ajustar los parámetros de acuerdo al problema tratado.

El presente trabajo tiene como objetivo la implementación de un algoritmo genético, el cual hará uso de diferentes técnicas de cruce autoadaptativa y se analizará el desempeño que tenga cada cruce con un banco de problemas ampliamente estudiados en la literatura. El análisis abarca la calidad de los resultados presentados y el tiempo de ejecución del algoritmo. También se añaden resultados para técnicas de cruce clásicas dentro del ámbito de los algoritmos genéticos, con el fin de poder comparar el desempeño de las técnicas autoadaptativas seleccionadas con respecto a las técnicas habitualmente utilizadas.

---

<sup>1</sup>Un problema NP (*No deterministic Polynomial*, por sus siglas en inglés) es un problema cuya solución se puede encontrar en tiempo polinomial mediante un algoritmo no-determinístico. Dado que tal algoritmo no existe, los problemas de tipo NP-difícil se caracterizan por tiempos de resolución que aumentan de forma exponencial con el tamaño de la instancia.



## 2. Justificación

Una desventaja frecuentemente mencionada de las técnicas metaheurísticas en general, y de los algoritmos evolutivos en particular, es la necesidad de ajustar numerosos parámetros. El ajuste adecuado de estos parámetros es, en muchos casos, crítico para el desempeño correcto del método, por lo cual un tema de investigación abierto es el desarrollo de mecanismos autoadaptativos, que permitan aprender del estado de la búsqueda para ajustar los parámetros al mismo tiempo que se busca una aproximación de la solución óptima.

Con respecto a la estrategia de cruza, la mayor parte de los trabajos publicados se refiere al ajuste autoadaptativo de la probabilidad de cruza y no tanto a la adaptación de la técnica de cruza en sí. En el marco del presente proyecto, se propone trabajar particularmente sobre técnicas de cruza autoadaptativas, que han sido estudiadas de forma relativamente escasa y dispersa en el tiempo. No se han encontrado, en la literatura especializada, trabajos que traten de evaluar y comparar su desempeño respectivo ante un conjunto de problemas, para poder decidir acerca de cuál estrategia autoadaptativa es mejor de acuerdo a las particularidades del problema abordado.

### 3. Antecedentes

Desde sus orígenes el hombre ha intentado explicar su procedencia. Durante muchos años, en diferentes sociedades y culturas, la tesis más aceptada sobre el origen del hombre y las especies que lo rodean fue el *creacionismo*; esta teoría sostiene que cada ser vivo que existe en el planeta fue creado por uno o varios seres divinos, colocando al ser humano en el punto más alto de la jerarquía de los seres vivos[1]. Gracias a la influencia que la iglesia católica y las religiones mayoritarias tuvieron durante siglos, el creacionismo se mantuvo como una tesis incuestionable y ampliamente aceptada. Fue varios años después de la Edad Media, durante la época de la Ilustración, que diferentes científicos propusieron teorías acerca del origen de las diferentes especies presentes en el planeta.

En el siglo XVIII, el naturalista y botánico francés George Louis Leclerc, Conde de Buffon, especuló que las especies se originaron entre sí, en una clara oposición al creacionismo. Basándose en las similitudes que apreció entre el hombre y los simios, Leclerc teorizó sobre un posible ancestro común entre ambas especies. Creía en los cambios en las especies a través del tiempo, pero no describió ningún mecanismo responsable de dichos cambios, sino que pensaba que era el ambiente el que influía directamente sobre los organismos.

A principios del siglo XIX, el biólogo francés Jean-Baptiste Lamarck propuso un mecanismo para explicar la transformación de los seres vivos a través de largos periodos de tiempo. Según la teoría de Lamarck, las características físicas adquiridas por un individuo serán transmitidas a sus descendientes, logrando así que las características que les permiten adaptarse mejor a su medio ambiente sean conservadas en las siguientes generaciones[2].

En 1859 el naturalista inglés Charles Darwin a través de su libro *El origen de las especies* postuló que la evolución de las especies se origina a través de cambios aleatorios de características hereditarias, combinadas con un proceso de *selección natural*. Lo verdaderamente revolucionario en la propuesta de Darwin fue el postular un mecanismo natural para explicar la génesis, diversidad y adaptación de los organismos[3]. La teoría propuesta por Darwin recibe el nombre de *darwinismo*.

En 1871 Darwin publicó su segundo libro, titulado *El origen del hombre*, en el cual aplica su teoría al origen de la especie humana. En su obra Darwin relacionó a los simios y al hombre en una misma línea evolutiva, lo cual le provocó un conjunto de críticas y burlas por parte de la sociedad (figura 1). Un caso muy conocido de las burlas que recibió Darwin es el del licor Anís del mono, que incluyó en la etiqueta una imagen de Darwin caricaturizado como simio llevando en la mano el lema “*Es el mejor, la ciencia lo dijo y yo no miento*”.

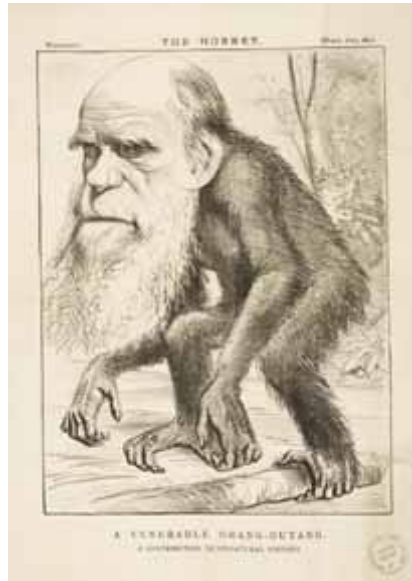


Figura 1: Caricatura de Darwin publicada por la revista Hornet en 1871.

Para 1866 el monje agustino Gregor Mendel publicó el resultado de los experimentos que realizó durante una década utilizando plantas de guisantes<sup>2</sup>. En su trabajo, Mendel demostró que los caracteres pasan de padres a hijos de manera predecible; de sus experimentos concluyó que los caracteres de los guisantes, como la altura de la planta y el color de la flor, están controlados por unidades hereditarias discretas que ahora llamamos *genes*. Además concluyó que los genes que controlan un carácter están en parejas, y que los miembros de cada pareja se separan en la formación de los gametos. Su trabajo constituye el fundamento de la *genética*, que se define como la rama de la biología que trata del estudio de la herencia y la variación[4]. Lamentablemente en el momento de su publicación los trabajos de Mendel no fueron valorados, sino hasta el año 1900 en que el biólogo alemán Carl Correns publicó los resultados de sus experimentos con plantas, en función de los trabajos de Mendel<sup>3</sup>.

A finales del siglo XIX, en 1892, el científico alemán August Weismann formuló una teoría conocida como *germoplasma*. Weismann sugirió la distinción entre el "plasma germinal", que se transmite de generación en generación, y el "plasma somático", que constituye el cuerpo de los organismos. Ambos factores serían independientes, de modo que cualquier modificación que sufriera el plasma somático no sería transmitida a los descendientes. Las características somáticas adquiridas por un individuo no serían heredables dado que — como las células que forman parte del cuerpo de los organismos son diferentes de las células sexuales (los óvulos y los espermatozoides) — los cambios en el cuerpo no pueden transferirse a las células germinales y, por lo tanto, no pasarían a la siguiente generación[5].

<sup>2</sup>Planta herbácea de la familia de las leguminosas, conocida en México como chícharo. Nombre científico: *Pisum Sativum*.

<sup>3</sup>El mismo año el botánico neerlandés Hugo de Vries publicó un trabajo similar al de Correns, pero sin citar la influencia de los trabajos de Mendel, lo que provocó la indignación de Correns y sus colaboradores.

Weismann fue un gran defensor de la selección natural, esta actitud era una consecuencia lógica de su teoría del plasma germinal o germoplasma. Como parte de las investigaciones sobre su teoría, realizó un experimento en el que cortó las colas de un grupo de ratas durante 22 generaciones<sup>4</sup>, sin que las nuevas generaciones de ratas perdieran dicha extremidad, contraponiéndose a la hipótesis fundamental de la teoría de Lamarck y demostrando que las adaptaciones somáticas no afectan al plasma germinal.

En 1896, el psicólogo estadounidense James Mark Baldwin publicó un trabajo, en la revista *American Naturalist*, en el cual propuso un nuevo factor en la evolución: la *selección orgánica*. Este nuevo factor consistía básicamente en la idea de que el aprendizaje es relevante desde una óptica evolutiva, en tanto que puede dirigir o guiar en alguna medida la evolución[6]. Baldwin planteó que si el aprendizaje ayuda a la supervivencia, entonces los organismos con mayor capacidad de aprendizaje tendrán más descendientes, incrementando de esa manera la frecuencia de los genes responsables del aprendizaje. Postuló un impacto indirecto del aprendizaje sobre el proceso evolutivo, y no uno directo como Lamarck. Actualmente su propuesta se conoce como *efecto Baldwin*.

En las décadas de los 30's y los 40's del siglo XX surge el *neodarwinismo* (también llamado teoría sintética de la evolución), el cual es básicamente el intento de fusionar el darwinismo clásico con la genética moderna, y fue formulado por científicos tales como George Gaylord Simpson, Ernst Mayr, Julian Huxley, Theodosius Dobzhansky, R. A. Fischer, Sewall Wright, y otros. Según esta teoría los fenómenos evolutivos se explican básicamente por medio de las mutaciones (las variaciones accidentales de que hablaba Darwin) sumadas a la acción de la selección natural. Así, la evolución se habría debido a la acumulación de pequeñas mutaciones favorables, preservadas por la selección natural, y por consiguiente la producción de nuevas especies no sería nada más que la extrapolación y magnificación de las variaciones que ocurren dentro de la especie[7].

El pensamiento evolutivo actual gira en torno al Neodarwinismo, estableciendo que toda la vida en el planeta puede ser explicada a través de sólo 4 procesos: reproducción, mutación, competencia y selección.

### 3.1. Historia de la computación evolutiva

A partir del año 1950 varios científicos de la computación estudiaron de forma independiente los sistemas evolutivos, con la idea de que la evolución podría ser utilizada como una herramienta de optimización para los problemas de ingeniería. El objetivo de todos estos sistemas era evolucionar una población de soluciones candidatas para un problema en específico, utilizando operadores inspirados por la variación genética natural y la selección natural[8].

El científico alemán Hans J. Bremerman es considerado uno de los pioneros más importantes de la computación evolutiva. Se considera que tal vez fue el primero en concebir a la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones en el campo de la computación evolutiva procesando cadenas binarias por medio de reproducción, selección y mutación. El trabajo de Bremermann se enfocó en el estudio y uso de los principios de la evolución

---

<sup>4</sup>El número total fue de 1592 ratas.

natural como mecanismos de optimización, siendo el primero en utilizar términos como población, genotipo, aptitud, selección y mutación.

En 1963 Ingo Rechenberg y Hans-Paul Schwefel, estudiantes de la Universidad Técnica de Berlín, se encontraron con la necesidad de resolver problemas de optimización mientras realizaban una serie de experimentos con un túnel de viento. Debido a que dichos problemas no podían resolverse analíticamente, Rechenberg desarrollo un método de ajustes discretos aleatorios inspirándose en el mecanismo de mutación que ocurre en la naturaleza. El método consistía en crear una mutación a partir de alguna solución y analizar el desempeño de la solución mutada, si esta nueva solución era mejor que su antecedente se utilizaba para crear nuevas mutaciones. Posteriormente Schwefel utilizó una computadora Z23 para implementar el método y probar la eficiencia del mismo. De esta manera nacieron las *estrategias evolutivas*[9].

En 1965 Lawrence J. Fogel, científico estadounidense, y sus colaboradores enfocaron su trabajo en la solución de problemas<sup>5</sup> mediante el uso de la evolución simulada. La técnica desarrollada — conocida como *Programación evolutiva* — se basa en la evolución de autómatas de estados finitos, los cuales eran sometidos a una serie de símbolos de entrada, esperando que al evolucionar fueran capaces de predecir los futuros símbolos de entrada. Para determinar el desempeño de predicción de un autómata se utiliza lo que Fogel nombró función de pago, mientras que para realizar la evolución de los autómatas se utiliza un operador de mutación. Es importante destacar que no se hace uso de ningún operador de recombinación, esto es debido a que Fogel pretendía modelar el proceso evolutivo al nivel de las especies y no al nivel de individuos [10].

Los *algoritmos genéticos* fueron inventados por el científico norteamericano John Henry Holland en la década de los 60's. A diferencia de las estrategias evolutivas y la programación evolutiva, la idea original de Holland no se centraba en resolver algún problema en específico; su propósito era el estudio formal del fenómeno de adaptación que se produce en la naturaleza y la forma en que estos mecanismos de adaptación pueden ser aplicados en los sistemas informáticos [8].

Durante los años siguientes Holland se dedico a desarrollar sus ideas con apoyo de alumnos y colegas de la Universidad de Michigan. Quince años después del inicio de las investigaciones de Holland, un estudiante llamado David Goldberg trató de aplicar los algoritmos genéticos para la solución de problemas industriales<sup>6</sup>. Holland consideró que la idea era sumamente complicada y trato de disuadir a su alumno, pero Goldberg no desistió y consiguió el objetivo auxiliado de una computadora Apple II. Esta y otras aplicaciones de los alumnos de Holland permitieron la difusión y generalización del estudio de los algoritmos genéticos en otras universidades.

Las estrategias evolutivas, la programación evolutiva y los algoritmos genéticos constituyen los tres paradigmas fundamentales de la computación evolutiva.

---

<sup>5</sup>El trabajo se desarrollo principalmente en problemas de predicción, control automático y teoría de juegos.

<sup>6</sup>Goldberg era ingeniero industrial y se dedicaba al diseño de pipelines.

## 4. Objetivos

### 4.1. General

Analizar y comparar diferentes técnicas de cruce adaptativa para algoritmos genéticos.

### 4.2. Específicos

1. Implementar dos versiones genéricas de un algoritmo genético: la primera con codificación binaria, la segunda con codificación real.
2. Implementar un conjunto de técnicas de cruce adaptativas diseñadas para trabajar con representaciones binarias (*Probabilistic Adaptive Crossover PAX*, *Statistics-based adaptive non-uniform crossover SANUX*, *Selective Crossover*).
3. Implementar un conjunto de técnicas de cruce adaptativas diseñadas para trabajar con representaciones reales (*Self-Adaptive Simulated Binary Crossover SBX*, *Genetic Parabolic Adaptive Crossover GPAX*, *Crossover Hill-Climbing XHC*).
4. Realizar un análisis del tiempo de ejecución y calidad del resultado para determinar el desempeño de cada técnica de cruce sobre algunas funciones de prueba reconocidas.

## 5. Marco teórico

Los algoritmos genéticos son un método adaptativo utilizado para resolver problemas de búsqueda y optimización. Su funcionamiento se basa en los principios de la teoría de la evolución de Darwin, la selección natural y la genética para generar soluciones a un problema determinado. Para cumplir con su objetivo, un algoritmo genético parte de una población inicial finita, donde cada individuo es una solución al problema. Los individuos son evaluados para medir su desempeño y a partir de los elementos más aptos se generan nuevos individuos mediante procesos de reproducción y mutación. Una vez generados los nuevos miembros de la población, se realiza un proceso de selección que tiene como objetivo aplicar una presión hacia las zonas prometedoras del espacio de búsqueda, además de mantener el tamaño de la población controlado y establecer los individuos que formaran parte de la siguiente generación<sup>7</sup>. Transcurrido un número predeterminado de generaciones se espera que los individuos originales de la población hayan evolucionado hacia la solución óptima del problema. Es importante mencionar que, como para cualquier técnica heurística, la optimalidad de las soluciones generadas no se puede garantizar.

### 5.1. Definición

David Goldberg presentó en 1989 su definición de algoritmo genético: “algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas”[11].

Tres años más tarde, el profesor e investigador de la Universidad de Standford, John R. Koza<sup>8</sup> presentó una definición bastante completa de algoritmo genético: “Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.”[12].

### 5.2. Estructura básica

La estructura básica de un algoritmo genético se puede describir mediante pseudocódigo:

```
BEGIN
  Generar la poblacion inicial
  Evaluar la aptitud de cada individuo
DO
  BEGIN
    Seleccionar individuos para la reproduccion
```

<sup>7</sup>Análogamente con la teoría darwinista, la selección natural permite que solo los mejores sobrevivan y que la población se adapte al medio ambiente

<sup>8</sup>Inventor de la Programación Genética.

```

Crear nuevos individuos mediante reproduccion
Mutar los nuevos individuos
Evaluar la aptitud de los nuevos individuos
Seleccionar a los miembros de la siguiente generacion
END
WHILE No se alcance el limite de generaciones
Presentar al individuo mas apto
END

```

De igual manera, se puede representar claramente como diagrama de flujo (Figura 2).

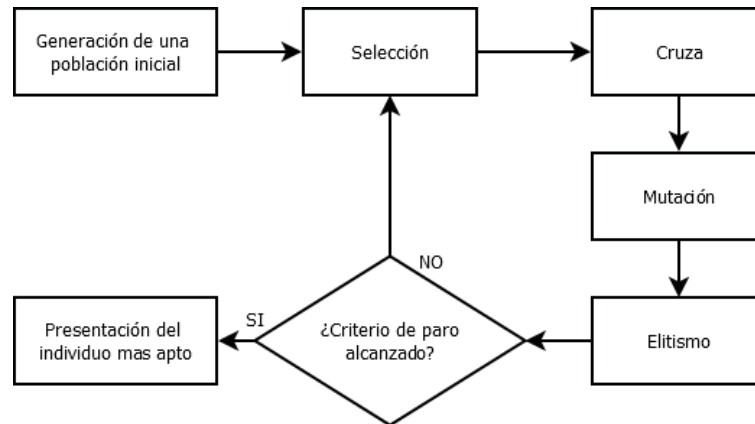


Figura 2: Estructura básica de un algoritmo genético simple.

### 5.3. Parámetros en un algoritmo genético

El modo operativo de un algoritmo genético involucra un grupo de parámetros que permiten controlar la forma en que interactúan los individuos de la población entre si. El resultado obtenido está ligado fuertemente a los valores que asignamos a estos parámetros, por lo que es importante su comprensión para poder realizar los ajustes correspondientes en pro de obtener una evolución satisfactoria.

Los principales parámetros son los siguientes:

**Número de generaciones ( $G$ ).** Indica cuantas generaciones se producirán a partir de la población original. Un número pequeño de generaciones puede limitar el proceso evolutivo de la población; sin embargo un número alto de generaciones no garantiza necesariamente la convergencia a la solución óptima, además de elevar el tiempo de procesamiento.

**Tamaño de la población ( $T_{pop}$ ).** Denota la cantidad de individuos que forman la población. En caso de que este valor sea insuficiente, las posibilidades de realizar reproducciones se reducen al disminuir el número de posibles combinaciones entre padres, provocando una búsqueda de soluciones escasa y poco óptima. En contraparte, una población excesiva permitirá una gran cantidad de reproducciones, sacrificando en velocidad al algoritmo.



**Tasa de supervivencia ( $T_s$ ).** Indica qué porcentaje de la población sobrevivirá para formar parte de la siguiente generación. Una tasa de supervivencia alta permite que el material genético de la población sufra menos alteraciones a través del paso de las generaciones, pero puede impactar en la convergencia del algoritmo.

**Tasa de recombinación ( $T_c$ ).** Indica en términos del porcentaje de la población el número de hijos que deben ser generados mediante la cruce de los padres. Usualmente se toma como  $1 - T_s$ , con el fin de generar los individuos necesarios para complementar a los supervivientes de cada generación.

**Tasa de mutación ( $T_m$ ).** Indica la probabilidad de que los individuos de la nueva población sufran alguna mutación en sus genes. Si la tasa de mutación es nula, la población es la misma después de la reproducción. Si la probabilidad es del 100 % toda la población se ve afectada por las mutaciones. Una tasa de mutación muy alta puede provocar que la población degenera muy rápidamente.

**Semilla de aleatorios.** La población inicial suele generarse mediante las funciones de números aleatorios propias del lenguaje de programación utilizado para implementar el algoritmo genético. Este tipo de funciones no generan realmente números aleatorios, sino que originan una serie de números pseudo-aleatorios a partir de un valor inicial conocido como *semilla*. El establecimiento de una semilla aleatoria permite la repetición de experimentos, al poder ejecutar el algoritmo genético nuevamente con la misma población inicial.

## 5.4. Codificación de individuos

La codificación se refiere a la forma en que los individuos de la población se representan en el algoritmo genético. Cualquier posible solución puede ser representada dando valor a un conjunto de parámetros. En la terminología de los algoritmos genéticos cada uno de estos parámetros es conocido como *gen*, y un conjunto de genes unidos en una cadena se denomina *cromosoma*. Un cromosoma en particular recibe el nombre de *genotipo*. El genotipo contiene la información necesaria para la construcción del *fenotipo* ( los rasgos físicos) de un individuo, es decir, las variables de una solución potencial del problema [13]. Descrito de otra manera, el genotipo es la cadena cromosómica utilizada para almacenar la información contenida en un individuo, mientras que el fenotipo se refiere a los valores que toman las variables tras *decodificar* el contenido del genotipo [10].

Como en cualquier método de búsqueda y aprendizaje, la forma en que las soluciones son codificadas es un factor central en el éxito del algoritmo genético. Tradicionalmente y en la versión canónica de los AGs se ha utilizado una cadena binaria de longitud fija. Sin embargo, en los últimos años se han realizado varios experimentos con otro tipo de codificaciones [8].

El esquema binario para la representación de soluciones se caracteriza por representar un cromosoma mediante una cadena de la forma  $\langle a_1, a_2, \dots, a_n \rangle$ , donde cada carácter de la cadena se denomina *alelo* y puede tomar como valor un uno ó cero [10]. Holland presentó en su libro [14]

una justificación teórica para usar codificaciones binarias. Realizó comparaciones entre dos representaciones: una mediante cadenas largas con pocos alelos y una de cadenas cortas con múltiples alelos. A partir de esto Holland argumentó que la primera codificación permite un número más alto de esquemas<sup>9</sup> que la segunda codificación, lo que favorece la probabilidad de aparición de buenos *bloques constructores*. Un bloque constructor se define como una porción de cromosoma que eleva la aptitud de la cadena a la que pertenece. De acuerdo a los resultados que obtuvo, Holland sostuvo que es preferible tener muchos genes con pocos alelos posibles que contar con una mayor cantidad de alelos en pocos genes. Esta teoría se justifica biológicamente al encontrar en la naturaleza cromosomas con muchas posiciones pero pocos alelos disponibles [14].

No obstante, si bien la codificación binaria es la que nos permite un mayor número de esquemas, también presenta una serie de desventajas al tratar de resolver problemas del mundo real. Las principales razones por las que una codificación binaria puede no ser la adecuada para resolver un problema dado son las siguientes:

- **Tamaño de las cadenas.** Si se desea representar variables reales mediante una codificación binaria y se necesita una buena precisión, el mapeo resultante necesitará cadenas binarias extremadamente largas, lo que impactará en la calidad de los resultados.
- **Epístasis.** El valor de un solo bit afecta de manera muy significativa las contribuciones de aptitud de otros bits en el genotipo.
- **Representación natural.** Algunos problemas (i. e. el problema de la mochila 0-1) se representan de manera excelente con una codificación binaria, en cambio, otros problemas no tienen una representación natural usando cadenas binarias (i. e. el problema del agente viajero hace uso de permutaciones de números enteros).
- **Ilegalidad.** En algunas ocasiones, la disposición de los bits puede generar una solución ilegal del problema.

Un problema de las codificaciones binarias que merece una mención aparte es que el uso de estas no mapea adecuadamente el espacio de búsqueda con el espacio de representación. Por ejemplo, en una cadena de 10 bits que representa números enteros, el número 511 se representa 011111111 y el número 512 se representa 100000000. De manera inmediata se puede notar que los 10 bits de la cadena sufrieron un cambio, mientras que los números enteros que representan solo variaron en una posición. Este fenómeno es conocido como *risco o distancia de Hamming*. Para solucionar este problema se usan representaciones alternativas que aseguren la propiedad de adyacencia en el espacio de representación. Los códigos de Gray<sup>10</sup> pertenecen a este grupo de representaciones alternativas.

---

<sup>9</sup>Un esquema es una plantilla que describe un subconjunto de cadenas que comparten ciertas similitudes en algunas posiciones a lo largo de su longitud [14].

<sup>10</sup>El código Gray, también llamado código binario reflejado, es un sistema de numeración binario en el que dos valores sucesivos difieren solamente en uno de sus dígitos.

Esta serie de problemas han llevado a la búsqueda de otras formas de codificación en los algoritmos genéticos. En la práctica se ha mostrado a través de un conjunto significativo de aplicaciones que el uso de números reales en un cromosoma proporciona mejores resultados que la codificación binaria. La desventaja es que la codificación mediante números reales ha implicado la re-formulación de los operadores genéticos clásicos, diseñados para cadenas binarias.

Una característica importante que presentan los algoritmos genéticos con codificación real es la de explotar la *gradualidad*. Este término hace referencia a los casos en los cuales un pequeño cambio en las variables se traduce en un cambio pequeño en la función [10]. Esta característica permite que un cambio pequeño en la representación sea mapeado como un cambio pequeño en el espacio de búsqueda.

## 5.5. Selección

La selección tiene la misión de determinar a los elementos de la población candidatos a reproducirse, así como los individuos a sobrevivir en la siguiente generación. El número de individuos a escoger depende directamente de la tasa de supervivencia  $T_s$ . El proceso de selección suele realizarse de forma probabilística, lo que permite que aún los elementos menos aptos de la población tengan una oportunidad de sobrevivir<sup>11</sup> [10].

Las técnicas de selección se clasifican de diversas maneras: proporcional, por torneo, estado uniforme, disruptiva, jerarquías no lineales, competitiva, etc. A continuación se describen los tipos de técnicas más usadas:

- **Selección proporcional.** Este tipo de técnica se distingue por elegir a los individuos de acuerdo a su contribución de aptitud con respecto al total de la población. Ejemplos de este tipo de técnicas son la Ruleta, Sobrante estocástico, Universal estocástico y Muestreo determinístico. El algoritmo de la ruleta es el más comúnmente usado desde los orígenes de los algoritmos genéticos; fue propuesto por De Jong [20] en 1975. Es una técnica muy sencilla, pero es muy ineficiente ( $O(n^2)$ ) con poblaciones muy grandes, además que no siempre respeta el número esperado de copias de un individuo.
- **Selección por torneo.** Este tipo de técnica utiliza comparaciones directas entre individuos. Se inicia mediante la distribución de los individuos sin un orden específico, después se seleccionan aleatoriamente  $n$  individuos (generalmente 2) y se escoge el más apto de todos. La selección por torneo es una técnica eficiente ( $O(n)$ ) y fácil de implementar. Suele presentar la desventaja de una presión de selección muy alta, dando muy poca esperanza de sobrevivir a los individuos menos aptos.

## 5.6. Operadores genéticos

Para pasar de una generación a otra y promover la evolución, se aplica sobre la población un conjunto de operadores genéticos. Ejemplos de estos operadores son los operadores de cruce,

<sup>11</sup>A diferencia de las estrategias evolutivas, donde la selección es extintiva y los elementos menos aptos tienen cero probabilidades de sobrevivir.

mutación y elitismo. Los detalles generales de estos operadores se describen a continuación:

**Cruza.** Una vez que han sido seleccionados los candidatos para la recombinación, se aplica una técnica de cruce para producir la descendencia que se insertará en la siguiente generación [13]. La técnica de cruce es la encargada de combinar el material genético de los padres elegidos para crear uno o varios individuos nuevos. Las técnicas básicas de cruce se desarrollaron enfocadas a la representación binaria, pero no son privativas de ésta, pues son adaptables a representaciones de mayor cardinalidad. Las técnicas básicas se describen a continuación:

- **Cruza de un punto.** Es una técnica propuesta por Holland [14]. Fue muy popular durante algunos años, pero no es muy usada en la práctica debido a sus desventajas (la principal de ellas es que hay esquemas a los que no se puede acceder desde esta técnica). Básicamente consiste en dividir el genotipo de ambos padres en un punto determinado de la cadena y generar dos hijos mediante la combinación de los segmentos de cadena resultantes (Figura 3). La ubicación del punto de ruptura de la cadena puede ser seleccionado aleatoriamente o ser determinado previamente y mantenerse fijo.



Figura 3: Ejemplo de cruce de un punto.

- **Cruza de dos puntos.** Se trata de una generalización de la cruce de un punto. En este caso se definen dos puntos de ruptura (Figura 4). Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora al cruce un punto, añadir mas puntos de ruptura promueve que se pierdan bloques constructores, sin embargo, promueve que el espacio de búsqueda del problema sea explorado con más profundidad. Ha sido demostrado que la cruce de dos puntos es la que más promueve la formación y preservación de bloques constructores.



Figura 4: Ejemplo de cruce de dos puntos.

- **Cruza uniforme.** Esta técnica, cada gen del nuevo individuo se obtiene por cualquiera de

sus padres, ya sea de manera aleatoria o mediante el uso de una máscara de bits<sup>12</sup>. Tiene un efecto disruptivo mayor que cualquiera de las dos cruza anteriores. Para controlar el efecto disruptivo se utiliza una probabilidad de cambio ( $P_c$ ) menor a 0,5.

En la práctica se suele realizar modificaciones a las técnicas de cruce con el objetivo de mejorar su desempeño. Existen diversas variantes estudiadas, por ejemplo, el retener sólo un hijo del par producido, restringir los puntos de cruce a aquellos en que los padres difieran (sustitución reducida), permutar partes de la cadena de los padres antes de la cruce (cruce con barajeo), establecer vectores de probabilidad adaptativa asociados a los padres (cruce adaptativa). Diferentes investigadores han presentado las ventajas y desventajas de utilizar este tipo de variantes.

Algunos investigadores han destacado que el énfasis de la cruce debe ser el poder generar todas las posibles combinaciones existentes en el espacio de búsqueda del problema. Mediante representación binaria, ni la cruce de un punto, ni la de  $n$  puntos son capaces de lograr este objetivo. Sin embargo, realizando ciertas variaciones a la cruce uniforme es posible lograrlo<sup>13</sup>.

En este punto es importante mencionar que al cambiar la representación, el operador de cruce requiere modificaciones a fin de mantener la integridad del nuevo esquema de codificación adoptado. En el caso de una representación real, el énfasis principal es la capacidad de romper un cierto valor real, de manera análoga a como la cruce ordinaria rompe segmentos de cromosoma al usarse representación binaria.

**Mutación.** La mutación consiste en modificar ciertos genes de forma aleatoria de acuerdo a la tasa de mutación establecida con anterioridad. Las mutaciones suelen en promedio ser beneficiosas pues contribuyen a la diversidad genética de la especie, además de prevenir que la población se vea limitada por un óptimo local. La mutación depende de la codificación. El abuso de la mutación equivale a reducir el comportamiento del algoritmo genético al de una simple búsqueda aleatoria, por lo que se recomienda el aumento del tamaño de la población o garantizar la aleatoriedad de la población inicial antes de aumentar la tasa de mutación [21].

Para el caso de una codificación binaria, la mutación consiste simplemente en la inversión del gen mutado que corresponderá con un bit. En el caso de una codificación real, es necesario reformular el operador de mutación. Las técnicas clásicas consisten en generar un nuevo valor del gen considerado, de acuerdo a una distribución de probabilidad con propiedades deseables, es decir, provocar una perturbación local y aleatoria.

**Elitismo.** Es el mecanismo mediante el cual el algoritmo genético asegura que los cromosomas de los miembros más aptos de la población pasen de generación en generación. El uso de este operador garantiza que la mejor aptitud en la población no se reducirá con el paso de las generaciones. Si bien el uso del elitismo no aumenta las posibilidades de acceder al óptimo global, se ha demostrado que el uso del elitismo es primordial para la convergencia del algoritmo.

---

<sup>12</sup>En caso de que el bit correspondiente de la máscara sea 1 se escogerá un determinado padre, en caso de que el bit sea 0 se escogerá al otro padre.

<sup>13</sup> Siguiendo esta línea, Radcliffe propuso una técnica denominada recombinación respetuosa aleatoria [22].

## 5.7. Ventajas y desventajas

En general, se pueden destacar las siguientes ventajas que presentan los algoritmos genéticos:

- Presentan simplicidad conceptual.
- Se pueden aplicar a una amplia variedad de problemas del mundo real.
- Presentan mejores resultados que algunas técnicas tradicionales en muchos problemas del mundo real.
- Son aptos para ejecutarse en paralelo.
- Generalmente pueden auto-adaptar sus parámetros.
- Son capaces de encontrar soluciones a problemas para los cuales no se conoce solución alguna.
- Se pueden combinar con otras técnicas de búsqueda.

Así mismo, presentan algunos inconvenientes:

- La representación del problema debe ser hecha mediante un lenguaje robusto que permita cambios aleatorios sin caer en errores fatales o resultados sin sentido.
- El tiempo de convergencia puede ser muy alto o simplemente el algoritmo puede no converger dependiendo de los parámetros iniciales.
- Se pueden presentar casos de convergencia prematura en diversas situaciones, principalmente cuando la población es pequeña y algún individuo se destaca, lo cual aumenta la probabilidad de que se merme la diversidad del material genético de la población.
- Como cualquier técnica metaheurística, no garantizan la optimalidad de la solución encontrada.

## 6. Desarrollo del proyecto

El trabajo desarrollado en este proyecto fue dividido en tres módulos, los cuales se muestran en la figura 5.

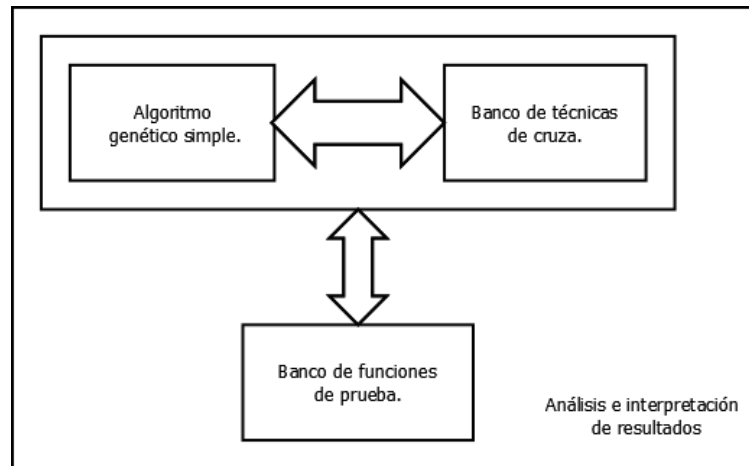


Figura 5: Descripción básica de los módulos del proyecto.

El algoritmo genético simple cuenta con la estructura mostrada en la figura 2. Se comienza con la generación de una población inicial de manera aleatoria. El objetivo de cada una de las siguientes etapas está bien definido: elegir los individuos de la población que tengan mayor grado de adaptación (*selección*), recombinar su material genético para producir nuevos individuos (*cruza*), alterar localmente las características de algunos de ellos para garantizar la diversidad (*mutación*) y seleccionar a los individuos que integrarán la siguiente generación (*elitismo*). La evolución de la población se detendrá hasta que se cumpla el criterio de paro establecido (número de generaciones).

El banco de técnicas de cruce se compone de una selección de técnicas tomadas de la literatura. Las técnicas de cruce adaptativa seleccionadas para codificación binaria son las siguientes:

- La técnica PAX (*Probabilistic Adaptive Crossover*), propuesta por Salah [23], se caracteriza por el uso de un vector de probabilidades adaptativo  $P$ , el cual representa la probabilidad de que la posición  $i$ -ésima de un hijo sea 1, en caso de que las posiciones  $i$ -ésimas de los padres sean diferentes. El vector se actualiza cada generación utilizando la información de la adaptación de los individuos de la población, de tal manera que los individuos que se encuentran por encima de la media de la población tengan una aportación más significativa mientras que la aportación de los individuos por debajo de la media sea castigada.
- La técnica SANUX (*Statistics-based adaptive non-uniform crossover*), propuesta en [24], utiliza la información de la frecuencia de los valores en cada locus para determinar en qué posiciones de los padres se realizarán intercambios de valores. Esta técnica utiliza un vector de frecuencias  $f_1$ , el cual indica la frecuencia de 1's en la posición  $i$ -ésima de la población. A partir de esta información se calcula el vector de probabilidad de intercambio  $p_s$  mediante

alguna de las funciones propuestas en [24] y de acuerdo a los valores obtenidos, se realiza el intercambio de valores sólo en las posiciones donde la probabilidad de intercambio es alta.

- La técnica *Selective Crossover*, tomada de [25], añade a cada miembro de la población un vector conocido como vector de dominancia, compuesto por valores reales entre 0 y 1. En cada recombinación, los cromosomas dominantes de ambos padres se transfieren a un hijo, y los restantes a otro. Posteriormente, ambos hijos son evaluados. Si la aptitud de un hijo es mayor que la de los padres, los valores del vector de dominancia son aumentados en aquellos cromosomas que fueron intercambiados durante la cruce. El valor en que son aumentados es proporcional al aumento en el desempeño del individuo.

Las técnicas de cruce adaptativa seleccionadas para codificación real se enlistan a continuación:

- La técnica SBX (*Simulated Binary Crossover*), propuesta en [26], tiene como objetivo emular el efecto de la cruce de un punto clásica para algoritmos genéticos binarios, para la codificación real. Utiliza una distribución de probabilidad que, de acuerdo a la posición de los padres, determina la posición de los dos hijos producidos. Esta función de probabilidad está diseñada para imitar la distribución implícita de los hijos resultantes de una cruce de un punto. Esta técnica es parametrizada por un factor  $\eta_c$  que determina la expansión de los hijos obtenidos (i.e., qué tan cerca quedan de sus padres). Posteriormente, se sugiere en [27] que este parámetro se autoadapte para permitir una exploración más fina del espacio de búsqueda, de acuerdo a la calidad de los hijos producidos con respecto a la de sus padres: si los hijos producidos son mejores que los padres, se modifica  $\eta_c$  para alejarlos aún más (se enfatiza la diversificación), mientras que, al contrario, si el hijo deteriora la calidad de los padres, se modifica  $\eta_c$  para encontrar más soluciones cercanas a las de los padres (se enfatiza el poder de intensificación). Los autores de [27] demuestran, mediante varios experimentos numéricos, que la cruce SBX autoadaptativa mejora significativamente los resultados obtenidos por la versión original.
- De la misma forma que la cruce SBX, la estrategia GPAX (*Genetic Parabolic Adaptive Crossover*, ver [28]) usa una distribución de probabilidad para determinar la ubicación de los hijos con respecto a la de sus padres. En este caso, la distribución de probabilidad es parabólica y parametrizada por dos factores,  $\alpha$  y  $\beta$ .  $\alpha$  y  $\beta$  pueden ser asociados, respectivamente, con un comportamiento promoviendo la exploración o la intensificación de la búsqueda. Los autores sugieren por lo tanto, al final de cada generación, asignar los valores de  $\alpha$  y  $\beta$  obtenidos por el mejor individuo actual a los individuos de la generación siguiente. Además, se aplican perturbaciones reducidas a los valores de  $\alpha$  y  $\beta$  para permitir ajustarlos dinámicamente.
- Como última cruce adaptativa para algoritmos genéticos con codificación real, se considera en el marco de este proyecto la técnica XHC (*Crossover Hill-Climbing*), inspirada, como lo indica su nombre, en estrategias de búsqueda local. Introducida en [29], esta técnica produce, para cada par de padres seleccionados, cierto número  $n_{off}$  de hijos mediante alguna técnica clásica. Si el mejor de estos hijos es, él mismo, mejor que el peor padre, lo sustituye. Este procedimiento es repetido cierto número de iteraciones. Para la técnica de cruce embebida



en este proceso, los autores de [29] sugieren usar el método  $BLX - \alpha$ , propuesto en [30], aunque mencionan que cualquier otra cruza para representación real podría igualmente ser elegida. Cabe mencionar que la cruza  $BLX - \alpha$  produce cada elemento del hijo muestreando uniformemente un intervalo incluyendo los dos padres, cuyo ancho es parametrizado por un parámetro  $\alpha$  típicamente igual a 0.5.

Cabe mencionar que se agregaron a este banco técnicas de cruza clásicas, no adaptativas, con el fin de tener una referencia en base a la cual comparar el desempeño de los seis métodos antes mencionados. Para codificación binaria, se agregó la técnica de cruza de 2 puntos, mientras que para codificación real se añadieron las técnicas no adaptativas SBX [26] y de recombinación intermedia.

Finalmente, en cuanto al módulo de Banco de Instancias, las funciones de prueba se eligieron de un conjunto de problemas ampliamente estudiados (función de Rosenbrock [15], función de Rastrigin [16], función esfera [20], función de Schwefel [17], función de Ackley [18] y función de Griewank [19]), esto con el fin de que existan resultados previos y sea factible realizar el análisis correspondiente y poder determinar la calidad de la solución obtenida. Las funciones seleccionadas se muestran a detalle en el Anexo A.

Una vez conformados ambos bancos, se realizaron experimentos numéricos para cada una de las funciones de prueba, variando la técnica de cruza ocupada por el algoritmo genético. En cada combinación posible, se calibró el algoritmo genético ejecutando el programa 28 veces, modificando los parámetros de entrada, pero manteniendo constante la semilla de los números aleatorios. Posteriormente, se revisaron los resultados para determinar el conjunto más adecuado de parámetros<sup>14</sup>, y nuevamente se ejecutó el programa 28 veces manteniendo constantes los parámetros previamente seleccionados, pero variando en cada corrida la semilla de los números aleatorios. A partir de los resultados obtenidos de estas últimas pruebas, se evaluó el desempeño de las técnicas de cruza estudiadas sobre las funciones de prueba propuestas.

## 6.1. Implementación

La implementación del algoritmo se realizó utilizando la biblioteca estándar del lenguaje de programación C++. Los parámetros que necesita el algoritmo genético para su funcionamiento los recibe el programa a través de la línea de comandos en el siguiente orden:

- Un número entero primo  $s$ , el cual es la semilla para la generación de números aleatorios.
- Un número entero  $T_{pop}$ , que representa la cantidad de individuos en la población.
- Un número entero  $G$ , que indica la cantidad máxima de generaciones sobre las que el algoritmo iterará.
- Un número de punto flotante  $T_s$ , que indica la tasa de supervivencia.
- Un número de punto flotante  $T_m$ , que representa la tasa de mutación.

<sup>14</sup>Para la selección de este conjunto de parámetros se hace uso de dos criterios: el mejor valor obtenido para la función objetivo y el promedio de los resultados obtenidos por la población.

- Un nombre de archivo, en el cual se guardarán los resultados de la evolución del algoritmo.

El algoritmo genético se realizó con la estructura planteada en la figura 2. Para la implementación de cada una de las técnicas de cruce se siguieron las descripciones correspondientes y algoritmos indicados por los autores de dichas técnicas. A continuación se enlistan detalles importantes que surgieron al momento de la implementación del algoritmo.

- La codificación binaria se realizó a través del contenedor *bitset* de C++, el cual optimiza el espacio utilizado por los bits, además de proporcionar métodos para la conversión directa a una variable entera e invertir el valor asociado a un bit.
- Para evitar el problema del riesgo de Hamming se utilizó el Código de Gray para la representación binaria.
- A cada cadena de bits se le asignó mediante mapeo un valor real con 6 decimales de precisión. El valor se asigna de la siguiente manera:

$$x = LI + \frac{n(LS - LI)}{2^l - 1} \quad (1)$$

Donde:

*LI* Limite inferior del rango de la variable

*LS* Limite superior

*n* Número entero que representa la cadena de bits

*l* Longitud de la cadena de bits

- La longitud de la cadena de bits depende del rango de la variable y la precisión en decimales que se requiera manejar. Se calcula de la siguiente forma:

$$l = 1 + \log_2((LS - LI)10^p) \quad (2)$$

Donde:

*l* Longitud de la cadena de bits

*LI* Limite inferior del rango de la variable

*LS* Limite superior

*p* Precisión de decimales utilizada

- La técnica SANUX indica construir una máscara a partir del vector de probabilidades calculado a partir de la frecuencia de 1's en los miembros de la población, pero no se especifica la forma de construir dicha máscara. Este detalle se solucionó por medio de una cota denominada probabilidad de cambio, determinada aleatoriamente. Si el vector de probabilidades en su posición *i*-ésima es mayor a la probabilidad de cambio, la posición *i*-ésima de la máscara será 1, en caso contrario será 0. Al realizar la cruce, los bits de los padres se intercambian en las posiciones donde la máscara vale 1.

- La técnica PAX mantiene actualizado cada generación un vector de probabilidades, el cual indica la probabilidad de que la posición  $i$ -ésima sea 1. Para determinar a partir de que probabilidad es aplicada esta regla se estableció una cota denominada probabilidad de cambio, determinada aleatoriamente. Si el valor del vector de probabilidades en su posición  $i$ -ésima es mayor a la probabilidad de cambio el valor en la posición  $i$ -ésima de la cadena será 1, en caso contrario será 0.
- La técnica GPAX muestra en su algoritmo que los parámetros  $\alpha$  y  $\beta$  pueden sufrir variaciones en  $\pm 0,1$ , pero no se indica un criterio para definir el signo de la variación, por lo que el signo de la variación se determino aleatoriamente. Otro detalle presente en la técnica GPAX es el número de veces que se debe aplicar la dicotomía para el calculo de la variable. En este caso se considero adecuado repetir el proceso 10 veces para obtener una buena precisión, sin comprometer el tiempo de ejecución.
- La técnica SBX Adaptativa indica que cada uno de lo dos hijos producidos en la cruce tendrá su correspondiente valor de  $\eta_c$ , pero no indica claramente un criterio para seleccionar cual de estos dos posibles valores se utilizará en la siguiente generación. En este caso se decidió utilizar el  $\eta_c$  del hijo mas apto.
- La técnica XHC indica la generación de un número  $n_{off}$  de hijos  $n_t$  veces, pero no se establecen los criterios de selección para estos valores. Se escogió un valor de 5 y 10 respectivamente para cada variable, considerando que estos valores no afectarán considerablemente el desempeño del algoritmo.

## 7. Resultados

### 7.1. Experimentos numéricos

En las tablas siguientes se recolectan los resultados obtenidos en los experimentos numéricos<sup>15</sup>. En cada tabla se muestra el mejor valor obtenido para la función objetivo, así como el valor de la desviación estándar de los resultados obtenidos en las pruebas.

#### 7.1.1. Codificación binaria

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	1.907360E-06	1.376070E-04
PAX	1.907360E-06	4.739539E-05
SANUX	1.907360E-06	2.314214E-04
Selective Crossover	1.788150E-06	7.476339E-05

Tabla 1: Resultados para la función de Ackley con codificación binaria.

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	2.104790E-11	1.182268E-06
PAX	9.313230E-13	4.814784E-09
SANUX	2.179300E-11	4.000000E-09
Selective Crossover	5.300000E-08	1.155662E-05

Tabla 2: Resultados para la función Esfera con codificación binaria.

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	6.196560E-05	8.896000E-03
PAX	2.127870E-05	9.617000E-03
SANUX	1.792640E-06	4.009000E-03
Selective Crossover	5.038980E-08	5.445000E-03

Tabla 3: Resultados para la función de Griewank con codificación binaria.

<sup>15</sup>En esta sección se muestran los resultados finales resumidos. En el apéndice B se muestran desglosados los resultados de todos los experimentos numéricos, así como los parámetros de entrada para cada una de las pruebas realizadas.

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	4.803960E-10	1.354790E-06
PAX	3.695180E-11	3.600000E-08
SANUX	1.847680E-10	3.643900E-07
Selective Crossover	3.695180E-11	2.088683E-06

Tabla 4: Resultados para la función de Rastringin con codificación binaria.

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	7.045490E-05	2.009200E-02
PAX	3.895200E-06	1.195000E-03
SANUX	7.080000E-07	1.624000E-03
Selective Crossover	3.904740E-06	8.833000E-03

Tabla 5: Resultados para la función de Rosenbrock con codificación binaria.

Cruza	Mejor resultado	Desviación estándar
Cruza de 2 puntos	2.545520E-05	2.173102E-06
PAX	2.545510E-05	1.322777E-07
SANUX	2.545520E-05	1.321501E-06
Selective Crossover	2.545590E-05	3.092855E-07

Tabla 6: Resultados para la función de Schwefel con codificación binaria.

### 7.1.2. Codificación real

Cruza	Mejor resultado	Desviación estándar
GPAX	-4.440890E-16	1.180400E-02
Recombinación intermedia	-4.440890E-16	9.945194E-04
SBX	-4.440890E-16	6.057323E-05
SBX Adaptativo	-4.440890E-16	1.280387E-06
XHC	-4.440890E-16	0

Tabla 7: Resultados para la función de Ackley con codificación real.

Cruza	Mejor resultado	Desviación estándar
GPAX	0	1.019470E-03
Recombinación intermedia	0	9.716335E-04
SBX	0	6.686900E-05
SBX Adaptativo	0	2.665189E-08
XHC	0	2.935139E-84

Tabla 8: Resultados para la función Esfera con codificación real.

Cruza	Mejor resultado	Desviación estándar
GPAX	0	1.062420E-03
Recombinación intermedia	0	2.638582E-02
SBX	0	2.907183E-02
SBX Adaptativo	0	2.342106E-05
XHC	0	0

Tabla 9: Resultados para la función de Griewank con codificación real.

Cruza	Mejor resultado	Desviación estándar
GPAX	0	8.527662E-02
Recombinación intermedia	0	7.993091E-02
SBX	0	1.614898E-02
SBX Adaptativo	0	2.375348E-06
XHC	0	0

Tabla 10: Resultados para la función de Rastrigin con codificación real.

Cruza	Mejor resultado	Desviación estándar
GPAX	0	8.527662E-02
Recombinación intermedia	0	7.993091E-02
SBX	0	1.614898E-02
SBX Adaptativo	0	2.375348E-06
XHC	0	0

Tabla 11: Resultados para la función de Rosenbrock con codificación real.

Cruza	Mejor resultado	Desviación estándar
GPAX	4.584800E-04	1.002873E-01
Recombinación intermedia	2.681480E-05	3.183470E-03
SBX	2.546090E-05	9.670000E-06
SBX Adaptativo	2.545510E-05	1.138034E+00
XHC	2.545510E-05	0

Tabla 12: Resultados para la función de Schwefel con codificación real.

## 7.2. Tiempo de ejecución

En las siguientes gráficas se muestra el tiempo de ejecución del algoritmo. Para realizar la medición de tiempo se ejecutó el algoritmo fijando la semilla aleatoria, el tamaño de la población, la tasa de supervivencia y la tasa de mutación; cambiando en cada experimento el número de generaciones que evoluciona la población.

### 7.2.1. Codificación binaria

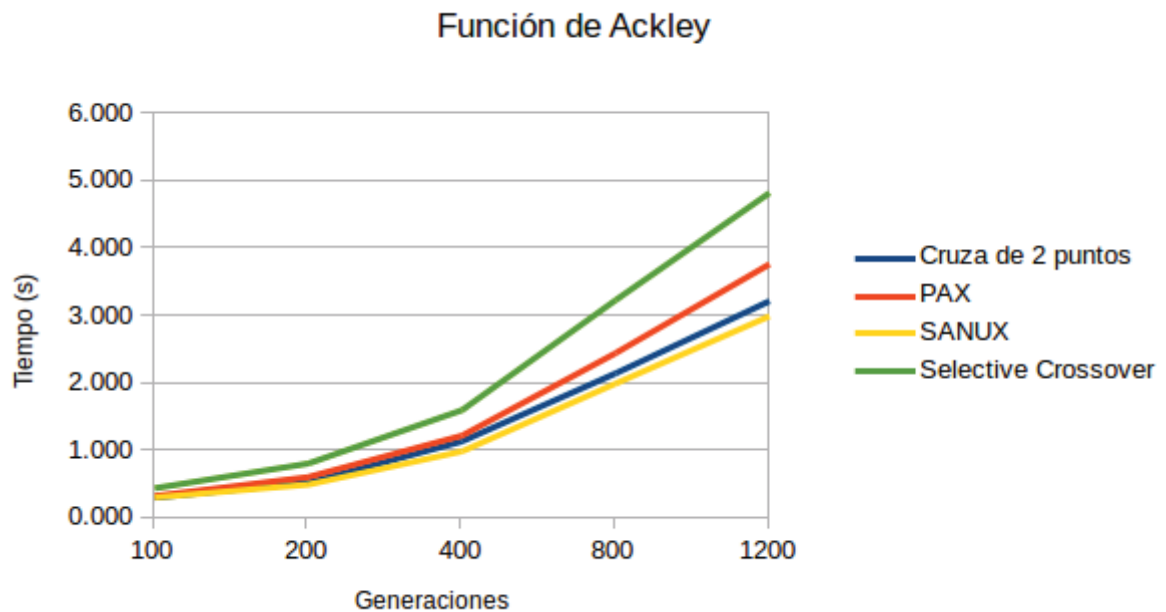


Figura 6: Tiempo de ejecución para la función de Ackley con codificación binaria.



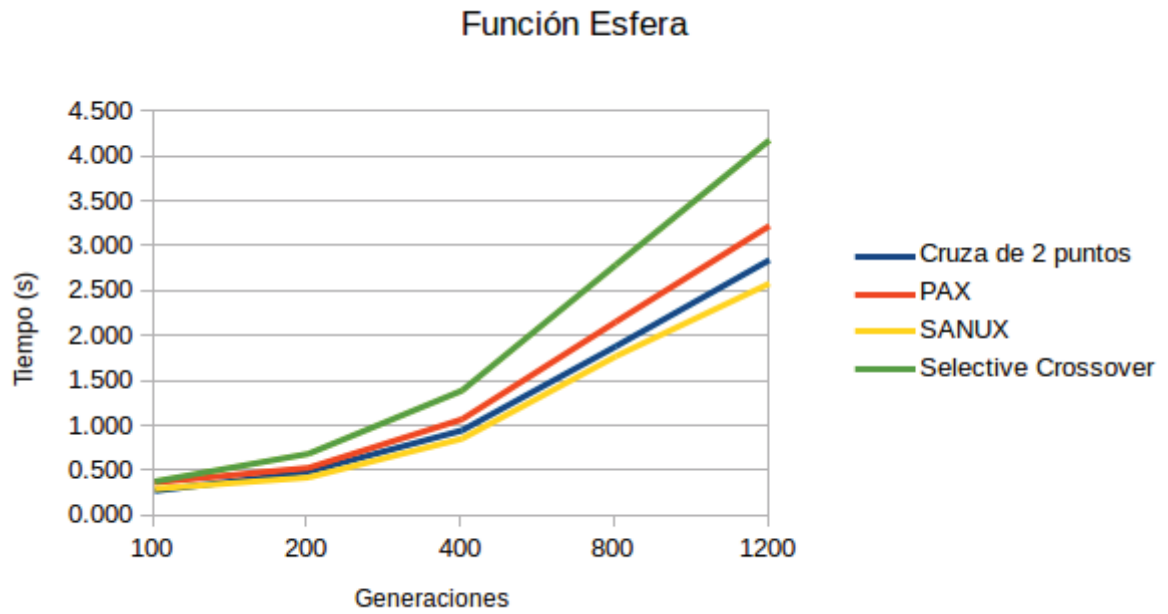


Figura 7: Tiempo de ejecución para la función Esfera con codificación binaria.

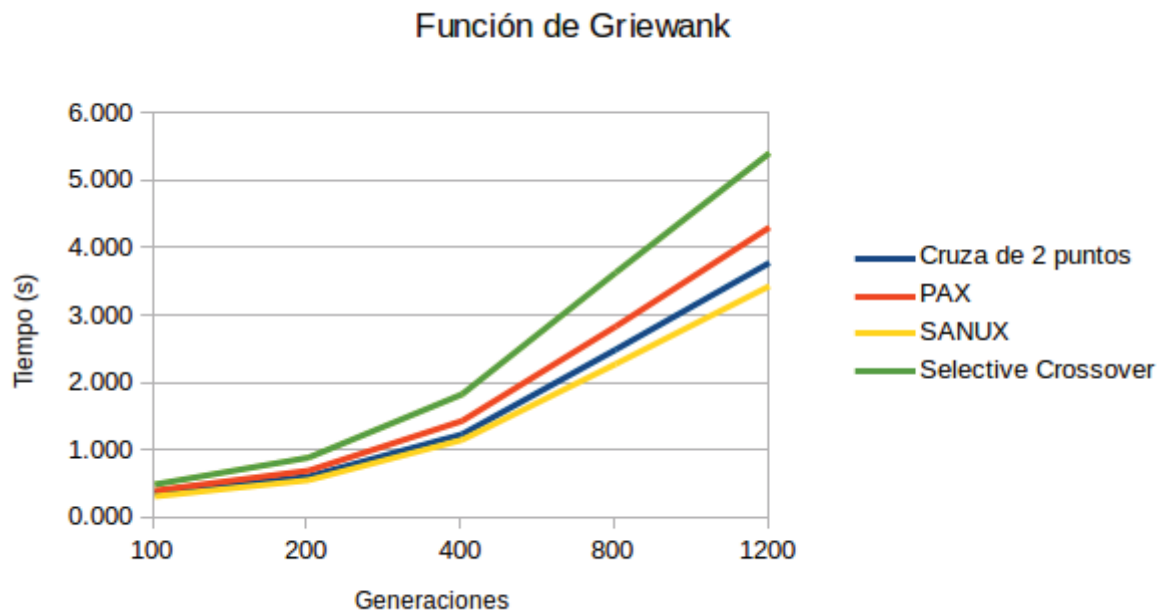


Figura 8: Tiempo de ejecución para la función de Griewank con codificación binaria.

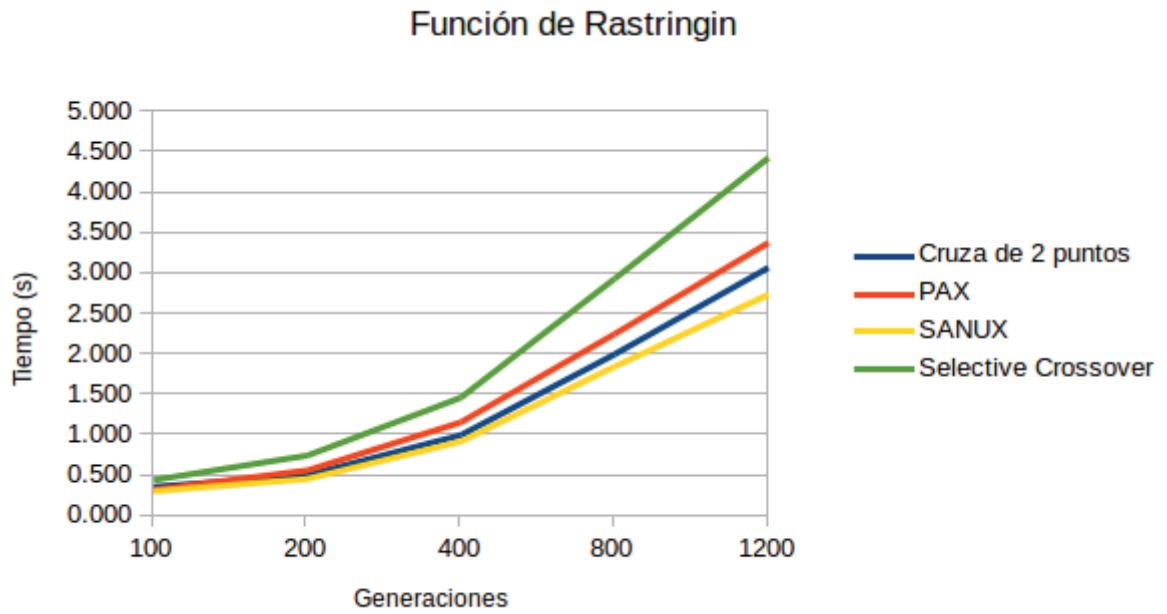


Figura 9: Tiempo de ejecución para la función de Rastrigin con codificación binaria.

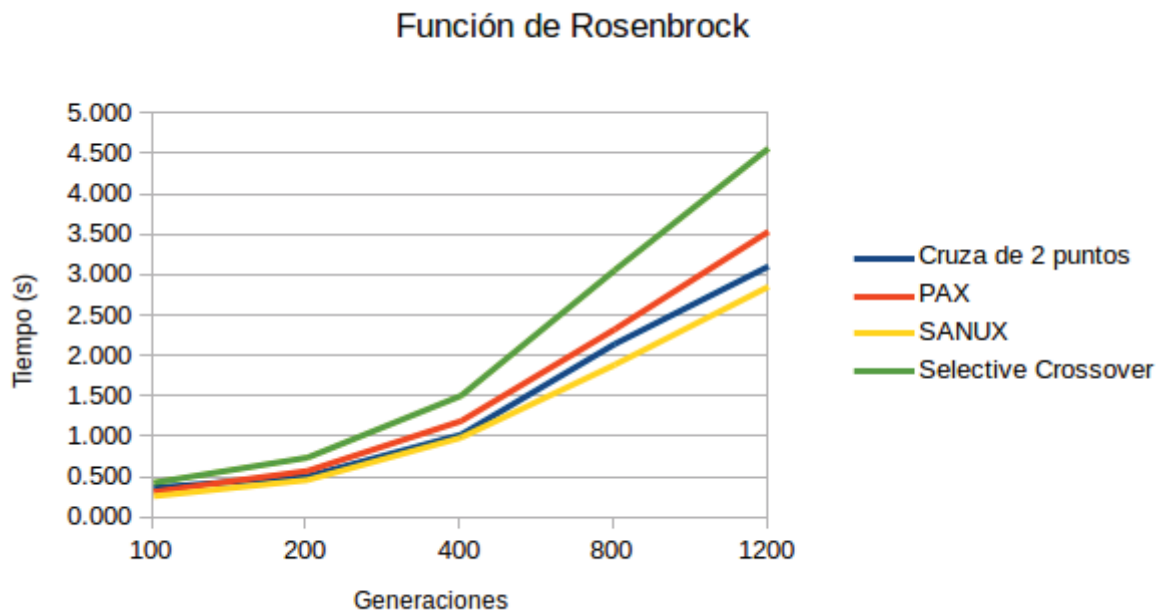


Figura 10: Tiempo de ejecución para la función de Rosenbrock con codificación binaria.

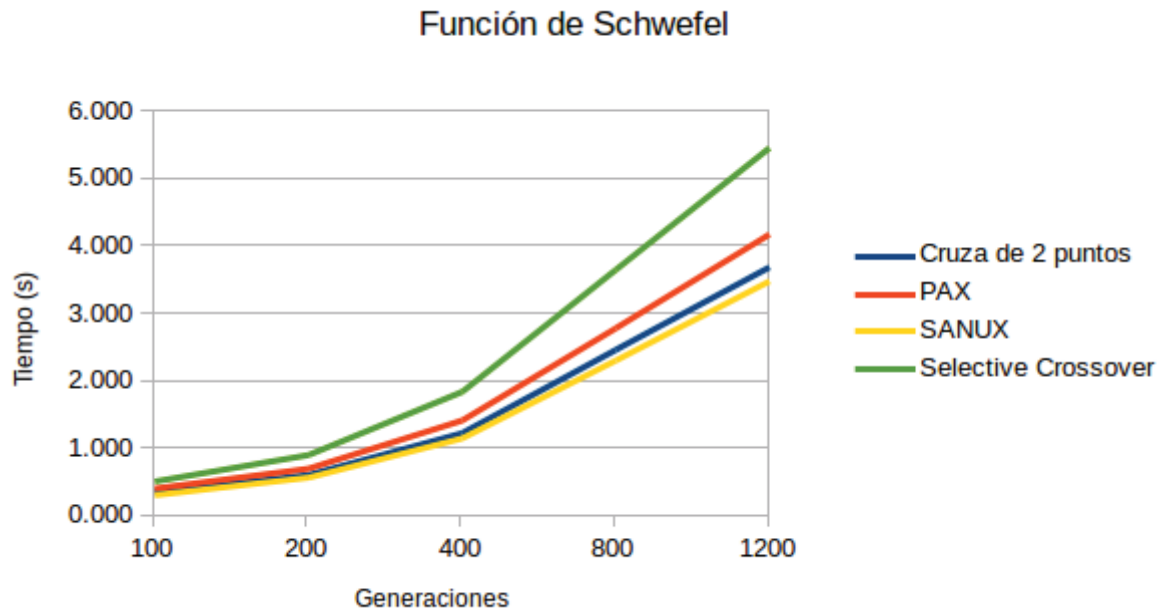


Figura 11: Tiempo de ejecución para la función de Schwefel con codificación binaria.

### 7.2.2. Codificación real

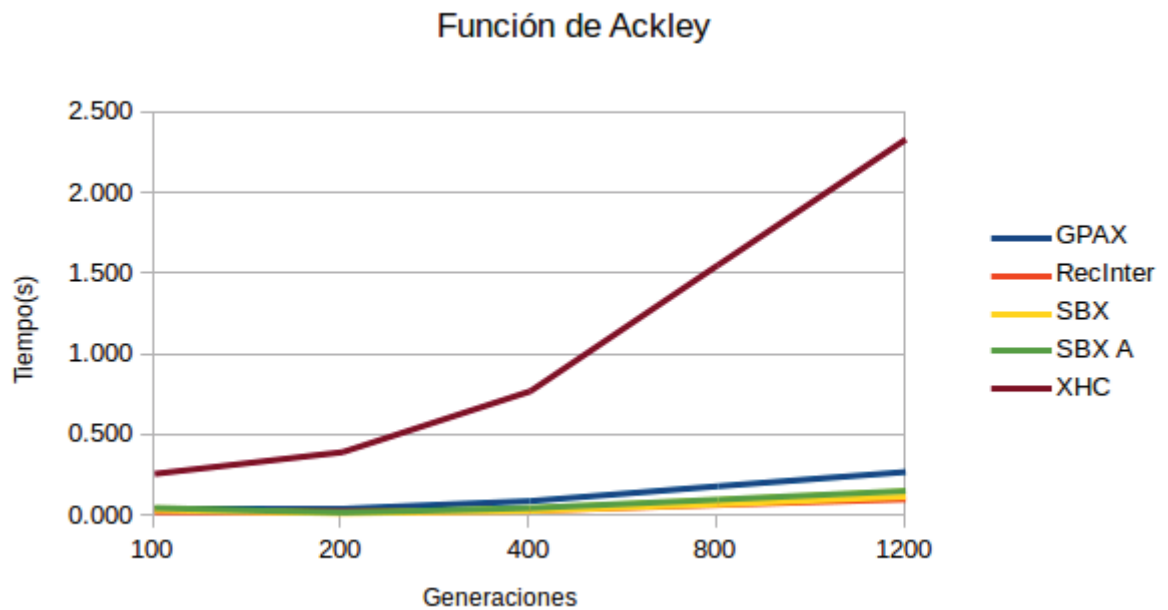


Figura 12: Tiempo de ejecución para la función de Ackley con codificación real.

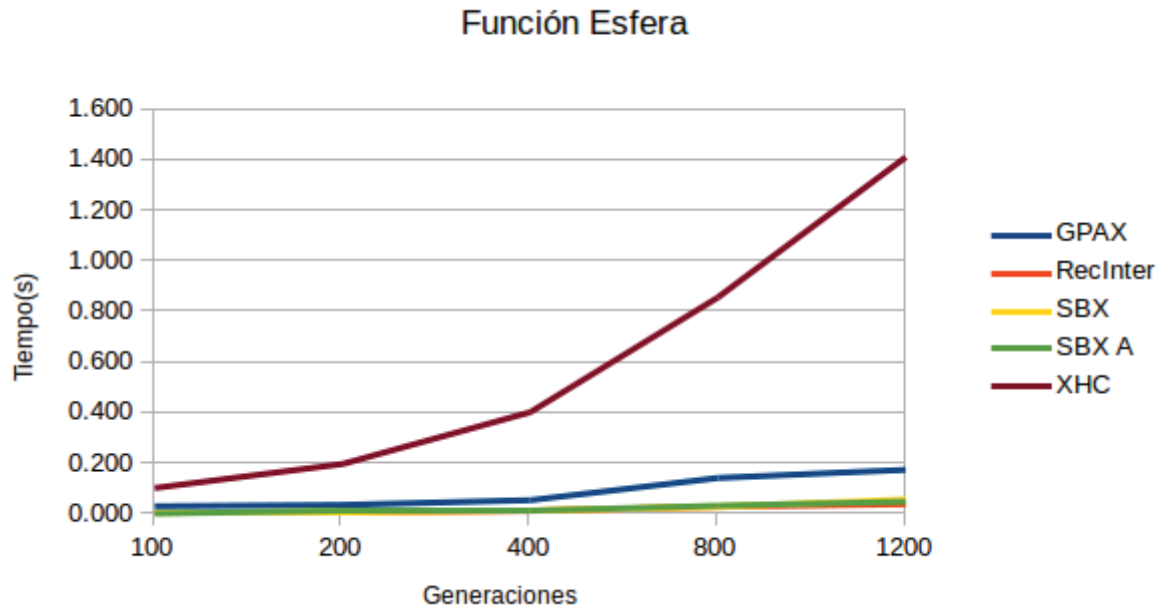


Figura 13: Tiempo de ejecución para la función Esfera con codificación real.

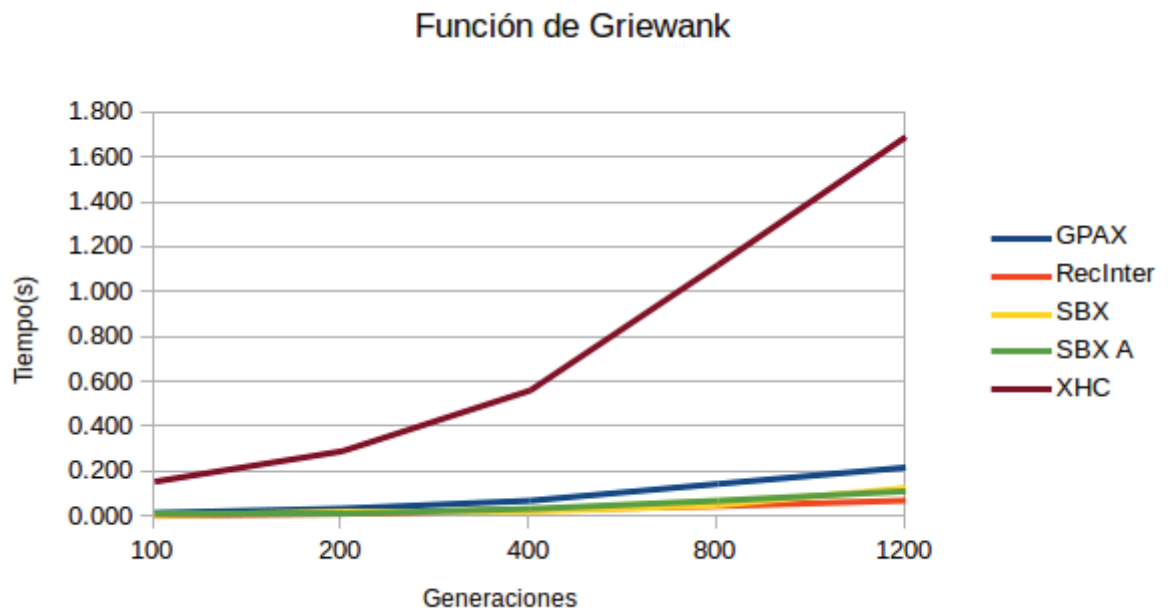


Figura 14: Tiempo de ejecución para la función de Griewank con codificación real.

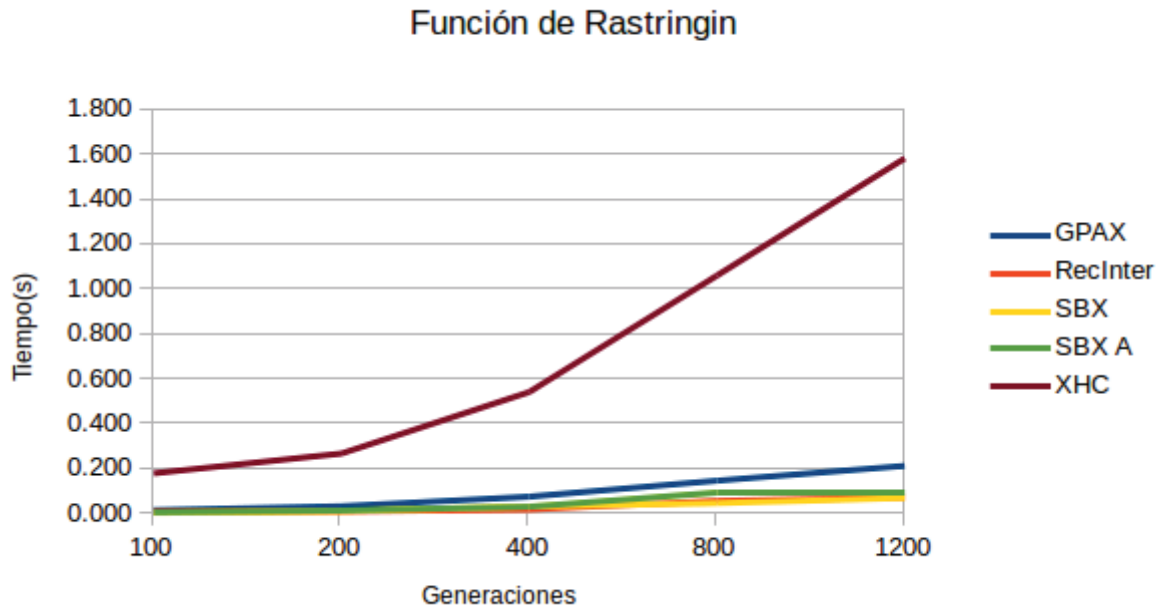


Figura 15: Tiempo de ejecución para la función de Rastrigin con codificación real.

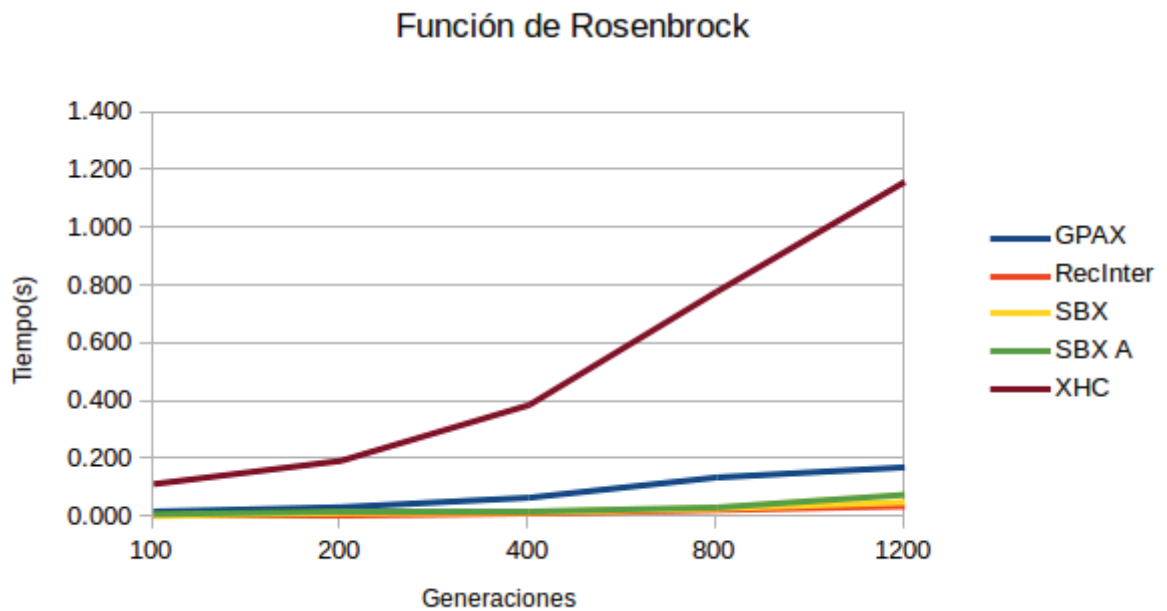


Figura 16: Tiempo de ejecución para la función de Rosenbrock con codificación real.

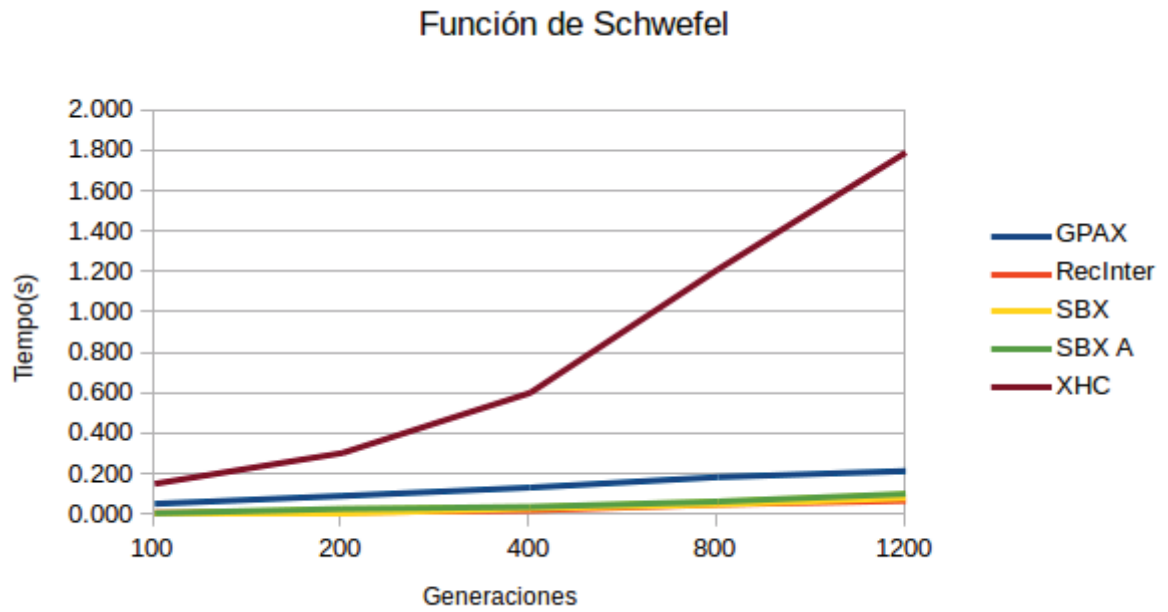


Figura 17: Tiempo de ejecución para la función de Schwefel con codificación real.

## 8. Análisis y discusión de resultados

Las pruebas realizadas con cada técnica de cruza han sido satisfactorias para cada una de las funciones de prueba. Aunque funciones como la de Griewank y la de Rastringin cuentan con una gran cantidad de óptimos locales (ver apéndice A), siempre se llegó a un resultado cercano al óptimo global.

En el caso del algoritmo genético con codificación binaria no es posible marcar una técnica como dominante en cuanto a la calidad de los resultados obtenidos. Los valores finales obtenidos para cada función se encuentran en un rango sumamente pequeño, obteniéndose una mínima diferencia entre estos valores. De igual manera los valores de la desviación estándar obtenidos se encuentran en un intervalo pequeño. Es importante mencionar que aunque los resultados obtenidos son muy cercanos a los óptimos globales de cada función, en ningún caso se llega a tocar el óptimo global.

En lo que respecta a los tiempos de ejecución, la técnica SANUX presenta el mejor tiempo de procesamiento en todos los casos, siendo la única técnica que logra una mejoría con respecto a la técnica clásica Cruza de 2 puntos. Al mismo tiempo es posible ver que la técnica Selective Crossover registra tiempos de ejecución notoriamente más altos que las otras tres técnicas, las cuales revelan curvas muy similares.

Fijando nuestra atención en el algoritmo genético con codificación real, los valores finales obtenidos para cada función son bastante satisfactorios. En los casos de la función Esfera, la función de Griewank, la función de Rastringin y la función de Rosenbrock siempre se llegó al óptimo global. En este punto se destaca la técnica de cruza XHC, la cual logra obtener los mejores valores de desviación estándar, incluyendo una desviación estándar nula, lo que indica que en todas las pruebas realizadas bajo esta técnica se logró llegar al mismo resultado.

Comparando la técnica SBX con su evolución, la técnica SBX Adaptativa, en ambos casos se presentan valores de la función objetivo bastante competitivos, diferenciándose únicamente en que la desviación estándar de los resultados de la técnica adaptativa mejoran con respecto a la técnica SBX clásica. La función de Schwefel fue la única en la que no se presentó esta mejora.

Analizando los tiempos de ejecución se puede observar que la técnica XHC presenta tiempos de procesamiento sumamente altos comparado con las otras 4 técnicas. Si bien es cierto que la técnica XHC mostró un desempeño superior en cuanto a la calidad de los resultados, el tiempo que necesita para presentar resultados puede llegar a ser inapropiado comparado con las demás técnicas de cruza. Por su parte, las técnicas SBX adaptativo y GPAX no mejoran el tiempo con respecto a las técnicas clásicas, pero la diferencia de tiempos es mínima.

El tiempo que tarda la técnica XHC es entendible debido al ciclo anidado que se encuentra en la parte central de la técnica. Los autores del método no limitan la forma o el rango en que se debe manejar el valor del número de iteraciones  $n_t$  ni el número de individuos  $n_{off}$  que se deben generar cada iteración, por lo que aumentar indiscriminadamente el valor de estos parámetros impacta directamente con el tiempo de ejecución.

## 9. Conclusiones

El trabajo de implementación y análisis realizado ha cumplido con el objetivo principal de este proyecto. Las pruebas realizadas sobre el conjunto de técnicas de cruce contra el banco de funciones de prueba nos permite verificar de qué manera impacta en el desempeño de un algoritmo genético la especialización de sus operadores, así como el tipo de codificación utilizada.

En las pruebas realizadas con codificación binaria no se logró llegar a ningún óptimo global. Esto no significa que el uso de esta codificación sea mala, sino que el mapeo existente entre las cadenas de bits y un rango de número reales presenta sesgo, lo que dificulta el acceso a ciertos valores del rango. En cambio, mediante el uso de codificación real se logró llegar a los óptimos globales de la mayoría de las funciones y en las funciones restantes se logró un mayor acercamiento al resultado deseado.

Ahora, hablando del tiempo de ejecución, el trabajo con codificación binaria es bastante lento con respecto a la codificación real. Esto último se debe a que en la codificación real cada individuo se compone de tantos valores numéricos reales requiera el problema, mientras que al trabajar con binarios debemos procesar una cadena de bits y además realizar el mapeo de valores correspondiente, el cual es un proceso muy largo pues implica convertir del código gray al código binario habitual, pasar a números enteros y calcular el respectivo valor dentro del intervalo de valores reales. Otro punto que se ve afectado al seleccionar el tipo de codificación es la implementación del algoritmo genético. El uso de codificación binaria implica el uso más tipos de estructuras de datos y métodos asociados para la transformación de cadenas en números enteros y mapeo de valores reales. También hay que mencionar que al momento de realizar la implementación, el uso de una técnica de cruce adaptativa requiere la inversión de un mayor tiempo de programación, además de que es necesario el cuidado y control de un número mayor de variables y parámetros; trabajo que puede ser no necesario si no se garantiza una mejora en el desempeño.

A pesar de las ventajas correspondientes al uso de codificaciones reales en algoritmos genéticos, en la práctica se encuentra muy difundido el uso de codificaciones binarias. En este sentido, queda abierta la elección de usar de técnicas de cruce adaptativa, debido a que los resultados que presentan son satisfactorios pero no hay una diferencia significativa con respecto a los resultados que se pueden obtener mediante las técnicas clásicas, que en general son más rápidas y consumen menos recursos para su ejecución.

Siguiendo este camino, aún hay bastante trabajo por realizar. Queda pendiente el diseño de técnicas de cruce adaptativa que nos permitan obtener resultados de calidad sin que se vean comprometidos el tiempo de procesamiento, ni la exploración y la explotación de los resultados. Un ejemplo de esto es la técnica XHC, la cual presenta excelentes resultados a cambio de sacrificar el tiempo de ejecución. A pesar de ser una técnica para codificación real, su tiempo de ejecución es sumamente parecido al tiempo tomado por un algoritmo genético con codificación binaria, lo que es bastante alto con respecto a las demás técnicas de cruce. .



## Referencias

- [1] A. Jiménez Martín. *Optimización multiobjetivo basada en Metaheurísticas*, Grupo de Análisis de Decisiones y Estadística, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid. 18 de Marzo de 2011. [En línea]. Available: [http://www.dia.fi.upm.es/~ajimenez/Docu\\_Metaheuristicas/Transparencias/Tema5Evolutivos.pdf](http://www.dia.fi.upm.es/~ajimenez/Docu_Metaheuristicas/Transparencias/Tema5Evolutivos.pdf). [Último acceso: 31 de Diciembre de 2013].
- [2] J. A. Castro Moreno, *De Lamarck a Darwin: ¿Continuidad o ruptura? ¿Linealidad o bifurcación?* Tecné, Episteme y Didaxis, no. 17, pp. 75-92, 2005.
- [3] A. Barbadilla, *LA SELECCIÓN NATURAL: "ME REPLICO, LUEGO EXISTO"*, Departamento de Genética y Microbiología. Universidad Autónoma de Barcelona, [En línea]. Available: <http://bioinformatica.uab.es/divulgacio/lasn/>. [Último acceso: 1 de Enero de 2014].
- [4] W. S. Klug, M. R. Cummings y C. A. Spencer, *Conceptos de Genética*, Madrid: Pearson Educación, S. A., 2006.
- [5] Curtis, Barnes, Schnek y Massarini, *Biología 7ma. Edición*, [En línea]. Available: <http://www.curtisbiologia.com/g1880>. [Último acceso: 1 de Enero de 2014].
- [6] V. M. Longa, *El efecto baldwin: su papel en biología evolutiva y su aplicación a la evolución del lenguaje*, LUDUS VITALIS, vol. XIII, nº 23, pp. 21-48, 2005.
- [7] B. Aceves Cano, *Neodarwinismo*, Universidad Autónoma de Guadalajara, [En línea]. Available: <http://genesis.uag.mx/edmedia/material/vidayev/tema06.cfm>. [Último acceso: 2 de Enero de 2014].
- [8] M. Mitchell, *An Introduction to Genetic Algorithms*, Massachusetts: The MIT Press, 1999.
- [9] A. Hernández Aguirre, C. A. Coello Coello y B. P. Buckles, *Estrategias Evolutivas: La versión alemana del Algoritmo Genético (Parte I)*, [En línea]. Available: <http://delta.cs.cinvestav.mx/~ccoello/compevol/estr1.pdf>. [Último acceso: 6 de Enero de 2014].
- [10] C. Coello Coello, *Introducción a la Computación Evolutiva (Notas de Curso)*, [En línea]. Available: <http://delta.cs.cinvestav.mx/~ccoello/compevol/apuntes.pdf>. [Último acceso: 10 de Enero de 2014].
- [11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Boston: Addison-Wesley Longman Publishing Co., 1989.
- [12] J. R. Koza, *Genetic Programming. On the Programming of Computers by Means of Natural Selection*, Massachusetts: MIT Press, 1992.

- [13] M. Gestal, D. Rivero, J. R. Rabuñal, J. Dorado y A. Pazos, *Introducción a los algoritmos genéticos y la programación genética*, Coruña: Servizo de Publicacións Universidade da Coruña, 2010.
- [14] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, Michigan: University of Michigan Press, 1975.
- [15] H. H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*, The Computer Journal, vol. 3, pp. 175-184, 1960
- [16] H. Mühlenbein, D. Schomisch y J. Born, *The Parallel Genetic Algorithm as Function Optimizer*, Parallel Computing, vol. 17, pp. 619-632, 1991.
- [17] H. Schwefel, *Numerical optimization of computer models*, Chichester, Wiley & Sons, 1981
- [18] D. H. Ackley, *A connectionist machine for genetic hillclimbing*, Boston, Kluwer Academic Publishers, 1987.
- [19] A. O. Griewank, *Generalized descent for global optimization*, Journal of Optimization Theory and Applications, vol. 34, n° 1, pp 11-39, 1981.
- [20] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems*, PhD thesis, University of Michigan, 1975.
- [21] J. Arranz de la Peña y A. Parra Truyol, *Algoritmos genéticos*, [En línea]. Available: <http://www.it.uc3m.es/~jvillena/irc/practicas/06-07/05.pdf>. [Último acceso: 27 de Enero de 2014].
- [22] Nicholas J. Radcliffe. *Forma Analysis and Random Respectful Recombination*. In Richard K. Belew and Lashon B. Booker, editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 222–229. Morgan Kaufmann Publishers, San Mateo, California, July 1991.
- [23] S. Salah Abuselme, *Cruza probabilística adaptativa, una nueva metodología de cruza para algoritmos genéticos*, Tesis para optar al grado de Magíster en Ciencias de la Ingeniería Mención Eléctrica, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Chile, 2004.
- [24] S. Yang, *Adaptive Crossover in Genetic Algorithms Using Statistics Mechanism*, in Artificial Life VIII, Standish, Abbass, Bedau (eds)(MIT Press), pp 182-185,2002.
- [25] K. Vekaria, C. Clack, *Selective Crossover in Genetic Algorithms: An Empirical Study*, Paper presented at the meeting of the PPSN, Lecture Notes in Computer Science, vol. 1498, pp. 438-447, 1998.
- [26] K. Deb, R. B. Agrawal. *Simulated Binary Crossover for Continuous Search Space*, Complex Systems, vol. 9, pp.115–148, 1995.

- [27] K. Deb, S. Karthick, T. Okabe, *Self-Adaptive Simulated Binary Crossover for Real-Parameter Optimization*, Proceedings of the Genetic and Evolutionary Computation Conference (GEC-CO'07), del 7 al 11 de julio 2007, Londres (UK), pp. 1187-1194, 2007.
- [28] J. L. Álvarez, M. E. Gegúndez, J. L. Arjona, *GPAX: Genetic Parabolic Adaptive Crossover*, Intelligent Automation and Soft Computing, vol. 18(3), pp. 225-236, 2012.
- [29] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, *Real-coded memetic algorithms with crossover hill-climbing*, Evolutionary Computation, vol. 12 no. 3, pp. 273-302, 2004.
- [30] L. J. Eshelman, J. D. Schaffer, *Real-coded genetic algorithms and interval-schemata*, In L. D. Whitley (editor), Foundations of Genetic Algorithms 2, Morgan Kaufmann, San Mateo, California, pp. 187-202, 1993.
- [31] J. E. Baker, *Reducing Bias and Inefficiency in the Selection Algorithm*, Proceedings of the Second International Conference on Genetic Algorithms and their Application (Hillsdale, New Jersey: L. Erlbaum Associates): pp. 14–21, 1987.

# Apéndices

## A. Funciones de prueba

A continuación se presentan en detalle las diferentes funciones de prueba utilizadas en el desarrollo de este proyecto.

### Función de Rosenbrock

Introducida por Howard H. Rosenbrock en [15], la función tiene un mínimo en el punto donde sus  $n$  variables valen 1 (Figura 18).

$$\begin{aligned} \text{mín } f(x) &= \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \\ & \quad -30 \leq x_i \leq 30 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \quad (3)$$

donde:  $f(x^*) = 0$  con  $[ x_1 \ \dots \ x_n ] = [ 1 \ \dots \ 1 ]$

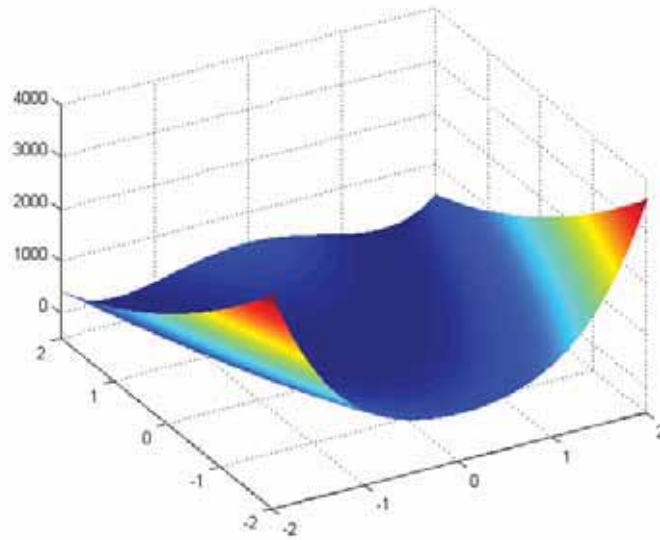


Figura 18: Gráfica de la función de Rosenbrock para dos variables.

## Función de Rastrigin

La función Rastrigin es una función no convexa utilizada para la realización de pruebas de rendimiento en algoritmos de optimización. Es un ejemplo típico de una función altamente multimodal<sup>16</sup>. Fue propuesta por primera vez por Rastrigin como una función de 2 dimensiones y fue generalizada por Mühlenbein [16]. Encontrar el mínimo de esta función es un problema bastante difícil debido a su amplio espacio de búsqueda y su gran número de mínimos locales (Figura 19).

$$\begin{aligned} \text{mín } f(x) &= \sum_{i=1}^n x_i^2 - 10 \cos(2\pi * x_i) + 10 \\ -5,12 &\leq x_i \leq 5,12 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \quad (4)$$

donde:  $f(x^*) = 0$  con  $[ x_1 \dots x_n ] = [ 0 \dots 0 ]$

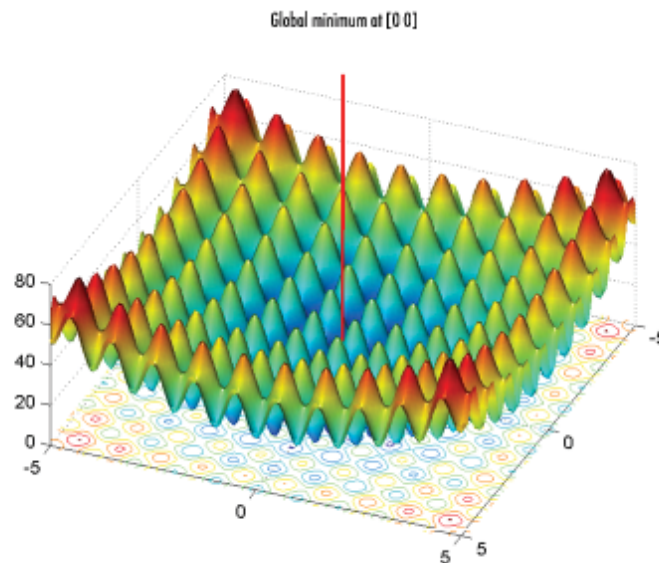


Figura 19: Gráfica de la función de Rastrigin para dos variables.

<sup>16</sup>Contiene varios máximos o mínimos dentro del espacio vectorial donde se define.

## Función esfera

La función esfera [20] es una sencilla función unimodal<sup>17</sup> (Figura 20).

$$\begin{aligned} \text{mín } f(x) &= \sum_{i=1}^n x_i^2 \\ -5,12 &\leq x_i \leq 5,12 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \tag{5}$$

donde:  $f(x^*) = 0$  con  $[ x_1 \ \dots \ x_n ] = [ 0 \ \dots \ 0 ]$

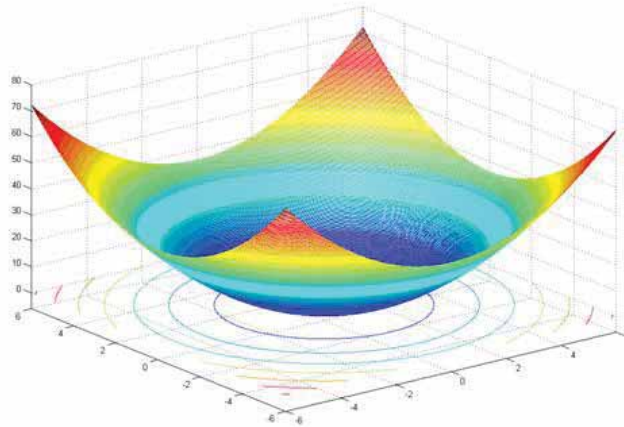


Figura 20: Gráfica de la función esfera para dos variables.

<sup>17</sup>Contiene un único mínimo o máximo global.

## Función de Schwefel

Es una función compleja, caracterizada por su gran cantidad de mínimos locales (Figura 21). El mínimo global es geoméricamente distante al mínimo local más próximo, por lo tanto, los algoritmos de búsqueda son potencialmente propensos a la convergencia hacia la dirección equivocada [17].

$$\begin{aligned} \min f(x) &= 418,9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) \\ &-500 \leq x_i \leq 500 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \quad (6)$$

donde:  $f(x^*) = 0$  con  $[ x_1 \dots x_n ] = [ 420,9687 \dots 420,9687 ]$

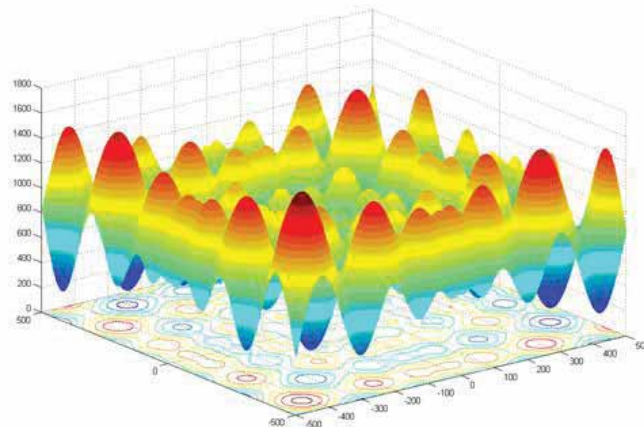


Figura 21: Gráfica de la función de Schwefel para dos variables.

## Función de Ackley

La función de Ackley [18] es ampliamente utilizada para las pruebas de algoritmos de optimización. En su forma de dos dimensiones (Figura 22), se caracteriza por una región externa casi plana, y un agujero grande en el centro. La función supone un riesgo para los algoritmos de optimización, que pueden ser atrapados en uno de sus muchos mínimos locales.

$$\begin{aligned} \text{mín } f(x) &= -20 \exp^{-0,2(\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^n})} - \exp^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} + 20 + \exp^1 \\ &\quad -32 \leq x_i \leq 32 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \quad (7)$$

donde:  $f(x^*) = 0$  con  $[ x_1 \dots x_n ] = [ 0 \dots 0 ]$

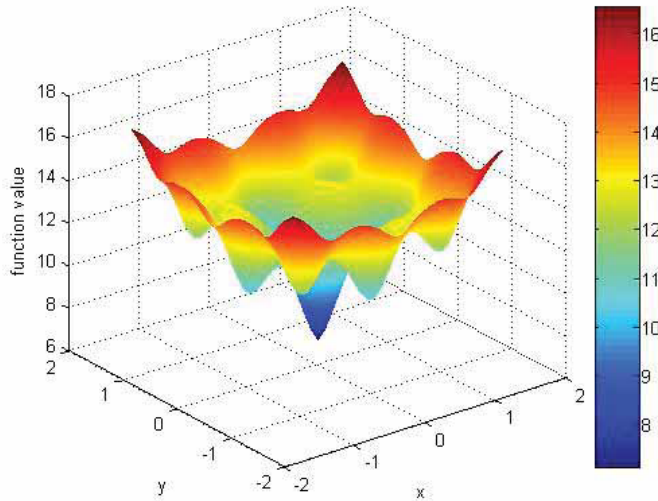


Figura 22: Gráfica de la función de Ackley para dos variables.



## Función de Griewank

La función de Griewank [19] contiene una gran cantidad de mínimos distribuidos (Figura 23)

$$\begin{aligned} \text{mín } f(x) &= \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ &-600 \leq x_i \leq 600 \text{ para todo } i = 1, 2, \dots, n \end{aligned} \quad (8)$$

donde:  $f(x^*) = 0$  con  $[ x_1 \dots x_n ] = [ 0 \dots 0 ]$

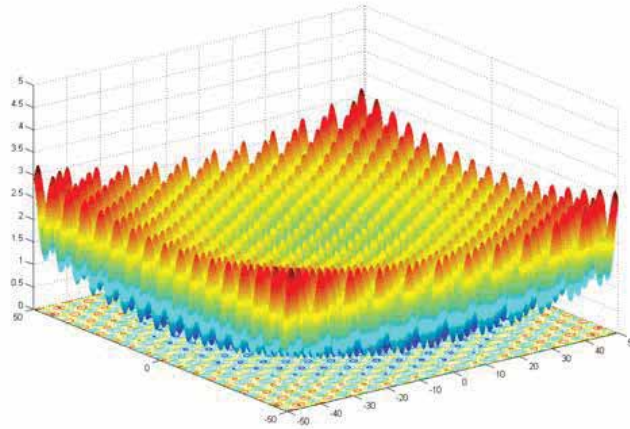


Figura 23: Gráfica de la función de Griewank para dos variables.

## B. Tablas de resultados

### B.1. Representación binaria

#### B.1.1. Ackley & Cruza de 2 puntos

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.072886	-0.0497956	0.446537	21.9596	0.893537
50	50	0.8	0.1	0.0414624	-0.0867991	0.502716	21.5591	0.135015
100	50	0.8	0.1	0.000700474	0.00130129	0.00423812	21.6876	0.20969
150	50	0.8	0.1	-0.0266786	-0.0664172	0.333783	21.9076	0.129736
150	150	0.8	0.1	-0.00223589	0.000215054	0.00648758	21.9493	0.0714464
150	150	0.7	0.15	0.00199747	-0.00259352	0.00954437	21.9624	0.103958
150	300	0.7	0.15	-5.96046E-005	0.00155783	0.00447414	21.945	0.132561
250	300	0.7	0.15	-0.000417233	0.000545025	0.00195396	21.9764	0.053183
350	300	0.7	0.15	-0.000505924	3.00407E-005	0.00144033	22.0314	0.0584769
400	300	0.7	0.15	-0.00120974	-0.00022459	0.00352043	22.1365	0.0497748
450	300	0.7	0.15	0.00100946	7.77245E-005	0.00289094	22.1302	0.0260203
500	300	0.7	0.15	-0.000388622	0.000132084	0.00116543	22.1316	0.0218268
500	400	0.7	0.15	-0.000385761	9.48906E-005	0.00112783	22.142	0.0380674
500	400	0.6	0.15	-7.77245E-005	1.57356E-005	0.000224465	22.113	0.0385798
500	400	0.5	0.15	-0.000199795	-6.53267E-005	0.000595722	22.1683	0.0224398
500	400	0.5	0.18	0.00895739	-0.00175142	0.0280318	22.1868	0.0178929
500	500	0.5	0.18	-0.00836229	-0.00188971	0.0262046	22.1139	0.0429489
600	500	0.6	0.18	-0.00269938	-8.44002E-005	0.00783293	22.2347	0.0303975
600	500	0.6	0.2	-0.00444746	0.000614643	0.0132356	22.1494	0.0297377
700	500	0.6	0.15	0.00136042	-0.000955105	0.00477503	22.1372	0.029156
700	500	0.65	0.15	-0.000145435	0.000592709	0.00173608	22.1007	0.0184653
750	600	0.65	0.15	-0.000310421	0.000099659	0.000924972	22.192	0.0260277
750	600	0.65	0.1	-0.000014782	0.000061512	0.000179042	22.2343	0.0242569
800	600	0.75	0.1	2.81334E-005	0.00015974	0.000459468	22.1551	0.00759218
800	600	0.75	0.07	1.85966E-005	4.76837E-007	5.26258E-005	22.0718	0.00979962
1000	1000	0.75	0.1	-7.15256E-006	3.33786E-006	2.23266E-005	22.1242	0.0174929
1000	1200	0.75	0.1	-5.24521E-006	-4.29153E-006	1.91698E-005	22.0828	0.0183049
1200	1000	0.75	0.1	1.43051E-006	1.38283E-005	3.93261E-005	22.0639	0.0119145

Tabla 13: Resultados para la función de Ackley utilizando Cruza de 2 puntos y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-4.76837E-007	-1.43051E-006	0.000004265	22.0976	0.0120137
19	-3.09944E-005	-6.19888E-006	8.94282E-005	21.9589	0.0103244
43	4.76837E-007	4.76837E-007	1.90736E-006	22.0959	0.0184825
83	-0.000014782	-3.33786E-006	4.28684E-005	22.0434	0.0140557
101	1.43051E-006	-4.76837E-007	0.000004265	22.1253	0.0232338
137	-1.85966E-005	-9.05991E-006	5.85207E-005	22.1333	0.00977609
151	2.38419E-006	-3.33786E-006	1.16024E-005	22.0801	0.00626592
191	-1.95503E-005	4.76837E-007	5.53233E-005	22.0778	0.0250268
211	-3.09944E-005	-0.000023365	0.000109825	22.0918	0.0103731
233	-8.10623E-006	8.10623E-006	3.24284E-005	22.0834	0.0237239
277	-1.57356E-005	-3.33786E-006	4.55043E-005	22.0253	0.0173647
313	2.14577E-005	-4.76837E-007	6.07187E-005	22.1153	0.0173773
349	-3.33786E-006	-4.76837E-007	9.53705E-006	22.0261	0.00487492
467	9.05991E-006	-4.76837E-007	2.56629E-005	22.0258	0.0176753
521	-7.15256E-006	1.19209E-005	3.93261E-005	22.1181	0.0115916
599	-4.29153E-006	-0.000023365	6.72068E-005	22.0792	0.0179119
617	7.15256E-006	8.10623E-006	3.05802E-005	22.0395	0.0095323
661	2.38419E-006	-1.57356E-005	4.50218E-005	22.0933	0.0264552
683	-4.76837E-007	-4.76837E-007	1.90736E-006	22.061	0.0225564
701	4.76837E-007	-1.43051E-006	0.000004265	22.0771	0.0272347
727	0.000114918	-4.76837E-007	0.000325391	22.1045	0.0111539
743	-0.000017643	-2.38419E-006	5.03639E-005	22.0836	0.00398905
797	1.43051E-006	4.29153E-006	1.27954E-005	22.1052	0.0183088
811	-2.38419E-006	-0.000246525	0.000698929	22.071	0.0133517
853	-0.000026226	-0.000014782	0.000085174	22.2175	0.0111814
883	4.76837E-007	-7.15256E-006	2.02768E-005	22.1515	0.0102423
907	-1.85966E-005	4.76837E-007	5.26258E-005	22.0421	0.0217649
991	1.43051E-006	4.76837E-007	0.000004265	22.1061	0.0259389

Tabla 14: Resultados para la función de Ackley utilizando Cruza de 2 puntos y una semilla de aleatorios variable.

Mejor resultado obtenido	1.90736E-006
Media de los resultados	7.107089E-005
Desviación estándar de los resultados	0.000137607

**B.1.2. Ackley & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.047338	0.0159774	0.206528	20.6955	0.838867
50	50	0.8	0.1	0.000409603	-0.0237641	0.0821976	21.5045	0.381086
100	50	0.8	0.1	0.018187	0.0030961	0.061219	22.0743	0.0357068
150	50	0.8	0.1	0.00141382	-0.0133376	0.042719	22.0258	0.120634
150	150	0.8	0.1	0.00019598	-0.00148058	0.00428363	21.9407	0.070766
150	150	0.7	0.15	0.00446844	-0.00101519	0.0135198	22.0561	0.0721559
150	300	0.7	0.15	0.00198317	-0.0018487	0.00786417	22.0079	0.0880608
250	300	0.7	0.15	0.00826406	-0.00105906	0.0254129	22.0832	0.0832104
350	300	0.7	0.15	0.00149202	0.0009799	0.00513368	22.1247	0.0407223
400	300	0.7	0.15	-0.00152445	-0.00199938	0.0072797	22.1633	0.0465606
450	300	0.7	0.15	0.00457907	-8.15392E-005	0.013512	22.1443	0.0460654
500	300	0.7	0.15	0.000586033	-0.00124502	0.00394248	22.145	0.0282557
500	400	0.7	0.15	0.00037241	-0.00112295	0.00338356	22.1726	0.0408982
500	400	0.6	0.15	-0.00309038	-0.000528812	0.00912971	22.0644	0.042978
500	400	0.5	0.15	-0.000796795	-0.00110769	0.00390897	22.2301	0.0162599
500	400	0.5	0.18	-0.000509739	-0.00326681	0.00964282	22.0971	0.0435159
500	500	0.5	0.18	-0.000509739	-0.00326681	0.00964282	22.094	0.0157385
600	500	0.6	0.18	0.000930309	-0.000719547	0.00336336	22.255	0.0296061
600	500	0.6	0.2	0.00266027	-0.000106335	0.00771914	22.2189	0.00455926
700	500	0.6	0.15	0.000105381	0.00132704	0.00381243	22.1319	0.0264146
700	500	0.65	0.15	0.00109625	-0.00111628	0.00449041	22.1697	0.0310766
750	600	0.65	0.15	0.000020504	0.000602245	0.00171406	22.1093	0.0277997
750	600	0.65	0.1	3.38554E-005	-1.85966E-005	0.000109293	22.1067	1.45724E-007
800	600	0.75	0.1	4.91142E-005	-9.29832E-005	0.000297725	22.1691	0.0130057
800	600	0.75	0.07	3.33786E-006	4.76837E-007	9.53705E-006	22.1849	0.00499586
1000	1000	0.75	0.1	3.38554E-005	0.000136852	0.000399275	22.1308	0.019982
1000	1200	0.75	0.1	3.09944E-005	-0.000136852	0.000397404	22.0837	0.00759958
1200	1000	0.75	0.1	-1.43051E-006	-3.33786E-006	1.02717E-005	22.1739	0.00340015

Tabla 15: Resultados para la función de Ackley utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	3.38554E-005	-4.29153E-006	9.65549E-005	22.0396	0.0183865
19	-4.76837E-007	1.09673E-005	3.10526E-005	22.1782	0.0177656
43	7.15256E-006	-7.15256E-006	0.000028613	22.1432	0.00957908
83	-4.76837E-007	4.76837E-007	1.90736E-006	22.0574	0.0221253
101	3.76701E-005	-2.38419E-006	0.000106798	22.0159	0.00503498
137	4.76837E-007	-1.28746E-005	3.64443E-005	22.0383	0.00831213
151	-4.76837E-007	5.24521E-006	1.48976E-005	22.1017	0.0192015
191	6.24657E-005	0.000017643	0.000183704	22.1097	0.026029
211	-4.76837E-007	1.66893E-005	4.72312E-005	22.1082	0.00594839
233	-2.38419E-006	4.76837E-007	6.8772E-006	22.1295	0.0086859
277	-4.76837E-007	1.43051E-006	0.000004265	22.0338	0.0184758
313	-2.38419E-006	2.38419E-006	9.53705E-006	22.106	0.0133729
349	-4.29153E-006	-8.10623E-006	0.000025945	22.0961	0.0259221
467	-8.10623E-006	-3.48091E-005	0.000101123	22.0889	0.0236905
521	1.43051E-006	4.76837E-007	0.000004265	22.0671	0.011555
599	-5.48363E-005	9.05991E-006	0.000157285	22.092	0.019982
617	1.43051E-006	2.38419E-006	7.86441E-006	22.1665	0.00871304
661	2.38419E-006	7.15256E-006	2.13263E-005	22.1047	0.0126023
683	-9.05991E-006	4.76837E-007	2.56629E-005	22.1076	0.0235239
701	1.57356E-005	-6.19888E-006	4.78437E-005	22.0055	0.0153532
727	-4.76837E-007	2.38419E-006	6.8772E-006	22.0748	0.00784958
743	3.33786E-006	-8.10623E-006	2.47976E-005	22.0789	0.0238036
797	-1.66893E-005	-1.19209E-005	5.80209E-005	22.1772	0.017678
811	-4.76837E-007	-4.76837E-007	1.90736E-006	22.0557	0.0122759
853	-4.76837E-007	1.43051E-006	0.000004265	22.068	0.00855433
883	0.000014782	1.43051E-006	4.20109E-005	22.1204	0.0074612
907	6.19888E-006	6.19888E-006	2.47976E-005	22.1159	0.00542291
991	-4.76837E-007	-4.76837E-007	1.90736E-006	22.0747	0.0239003

Tabla 16: Resultados para la función de Ackley utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	1.90736E-006
Media de los resultados	4.013505E-005
Desviación estándar de los resultados	4.739539E-005

**B.1.3. Ackley & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.218963	-0.0575929	1.58995	19.8732	0.924437
50	50	0.8	0.1	0.00669432	0.029655	0.110418	21.768	0.150672
100	50	0.8	0.1	0.0390859	-0.0203347	0.175598	22.0271	0.120232
150	50	0.8	0.1	0.00446558	-0.0104661	0.0356291	21.9217	0.117214
150	150	0.8	0.1	-0.00106573	0.000761509	0.00375047	21.7519	0.141381
150	150	0.7	0.15	0.0137067	0.00415754	0.0459669	21.9864	0.144902
150	300	0.7	0.15	-0.00785494	0.000449657	0.023901	21.9947	0.0929309
250	300	0.7	0.15	-1.38283E-005	-0.00246763	0.00714177	22.0985	0.0877014
350	300	0.7	0.15	-0.00119638	-0.00114107	0.004749	22.0745	0.0583844
400	300	0.7	0.15	-0.00198603	-0.00251246	0.00933146	22.202	0.0390709
450	300	0.7	0.15	-0.00598764	-0.000302792	0.0179141	22.1907	0.00926269
500	300	0.7	0.15	0.00114393	-0.00788736	0.0242328	22.2466	0.0398698
500	400	0.7	0.15	-0.0011754	6.91414E-005	0.00336721	22.1072	0.0306066
500	400	0.6	0.15	-0.00140238	0.000524044	0.0042941	22.1965	0.0107747
500	400	0.5	0.15	-0.000426769	0.00296545	0.00871297	22.2198	0.0107852
500	400	0.5	0.18	-0.00294447	0.00176001	0.0100159	22.1799	0.0249499
500	500	0.5	0.18	-0.00294447	0.00176001	0.0100159	22.1928	0.0424605
600	500	0.6	0.18	-0.00200415	0.000359058	0.00586922	22.0965	0.010436
600	500	0.6	0.2	-0.000897884	-0.00160456	0.00529064	22.1398	0.0215365
700	500	0.6	0.15	-0.000380039	0.000658512	0.00216587	22.1184	0.0274856
700	500	0.65	0.15	4.14848E-005	-0.00215673	0.0062252	22.1196	0.0294674
750	600	0.65	0.15	0.00152445	0.000734806	0.00486281	22.1188	0.0267024
750	600	0.65	0.1	-0.000342846	-8.10623E-006	0.000973117	22.1274	0.015422
800	600	0.75	0.1	3.29018E-005	3.95775E-005	0.000145643	22.0302	0.0260089
800	600	0.75	0.07	-9.05991E-006	9.05991E-006	0.000036244	22.1414	0.0248836
1000	1000	0.75	0.1	3.19481E-005	-3.86238E-005	0.000141841	22.106	0.0155048
1000	1200	0.75	0.1	3.09944E-005	2.24113E-005	0.000108221	22.0833	0.0208981
1200	1000	0.75	0.1	-4.76837E-007	1.43051E-006	0.000004265	22.1444	0.00856278

Tabla 17: Resultados para la función de Ackley utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1200), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	3.33786E-006	-4.76837E-007	9.53705E-006	22.1187	0.0138092
19	-4.76837E-007	4.72069E-005	0.000133587	22.1016	0.0112627
43	-4.29153E-006	-4.76837E-007	1.22135E-005	22.104	0.0175723
83	-3.38554E-005	2.38419E-006	9.60255E-005	22.1555	0.0176579
101	0.000017643	2.14577E-005	7.85931E-005	22.1188	0.0164637
137	2.38419E-006	3.09944E-005	8.79502E-005	22.1106	0.0169828
151	4.29153E-006	-4.29153E-006	1.71671E-005	22.1239	0.0157627
191	4.29153E-006	4.76837E-007	1.22135E-005	22.1169	0.013254
211	7.15256E-006	-1.19209E-005	3.93261E-005	22.2558	0.00861988
233	3.09944E-005	-0.000257015	0.000734	22.0713	0.0171954
277	2.24113E-005	-2.71797E-005	9.96726E-005	22.0891	0.0112485
313	1.43051E-006	-0.000017643	0.000050074	22.1671	0.0072284
349	-8.10623E-006	-1.09673E-005	3.85787E-005	22.1999	0.0177712
467	-9.48906E-005	-3.00407E-005	0.000281784	22.1582	0.0118642
521	6.19888E-006	6.24657E-005	0.000177652	22.0957	0.0054435
599	-3.33786E-006	-1.43051E-006	1.02717E-005	22.1091	0.016637
617	4.76837E-007	9.05991E-006	2.56629E-005	22.1873	0.0159982
661	-2.38419E-006	-1.66893E-005	4.76913E-005	22.1777	0.0178289
683	5.24521E-006	8.10623E-006	2.73116E-005	22.1944	0.0110031
701	1.00136E-005	4.76837E-007	2.83575E-005	22.111	0.0162511
727	-1.43051E-006	6.19888E-006	0.000017995	22.094	0.0113034
743	-7.15256E-006	-7.15256E-006	0.000028613	22.101	0.0173695
797	9.05991E-006	8.10623E-006	3.43892E-005	22.1562	0.0140353
811	-0.000017643	3.09944E-005	0.000100907	22.1252	0.0120137
853	-1.85966E-005	-1.38283E-005	6.55616E-005	22.1501	0.00406176
883	0.000014782	-3.33786E-006	4.28684E-005	22.1747	0.0119316
907	-4.76837E-007	4.76837E-007	1.90736E-006	22.1652	0.00634577
991	-3.95775E-005	0.000370502	0.0010576	22.0846	0.010122

Tabla 18: Resultados para la función de Ackley utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	1.90736E-006
Media de los resultados	0.0001199111
Desviación estándar de los resultados	0.0002314214

**B.1.4. Ackley & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.992344	4.33624E-005	2.62029	20.7585	0.880116
50	50	0.8	0.1	0.00945792	0.0409524	0.165266	21.832	0.202245
100	50	0.8	0.1	-0.000746101	0.00495896	0.0148534	21.8569	0.174333
150	50	0.8	0.1	-0.00752494	-0.000363439	0.0228192	22.0132	0.11084
150	150	0.8	0.1	-0.000228435	-0.00161603	0.0046872	21.8527	0.112647
150	150	0.7	0.15	-0.00565186	0.00389233	0.0206637	21.9042	0.126993
150	300	0.7	0.15	-0.00160262	-0.000753254	0.00509212	21.9348	0.0967939
250	300	0.7	0.15	-0.000503808	-0.00022307	0.0015665	22.032	0.0167711
350	300	0.7	0.15	-0.000505596	0.00057444	0.00218005	22.043	0.0626762
400	300	0.7	0.15	0.00066027	0.0020577	0.0062367	22.1044	0.0105222
450	300	0.7	0.15	0.00346676	-0.000565499	0.0102636	22.0365	0.0331393
500	300	0.7	0.15	-0.000531524	-0.000919551	0.00303416	22.0368	0.0171772
500	400	0.7	0.15	-0.000531524	-0.000919551	0.00303416	22.142	0.0270957
500	400	0.6	0.15	0.000204295	-0.000159591	0.000735034	22.067	0.0344704
500	400	0.5	0.15	-0.00122711	0.000461787	0.0037542	22.036	0.0305584
500	400	0.5	0.18	0.000207871	0.00100806	0.00293943	22.0169	0.0231798
500	500	0.5	0.18	0.000206977	0.0010027	0.00292377	22.2076	0.0430714
600	500	0.6	0.18	0.000351816	-0.00386819	0.0113877	22.1483	0.0341163
600	500	0.6	0.2	0.000701398	0.000769347	0.00297349	22.1603	0.0352468
700	500	0.6	0.15	-6.21378E-005	0.000127405	0.000401465	22.0718	0.0310918
700	500	0.65	0.15	-0.000475198	0.000012964	0.00135058	22.1329	0.0301766
750	600	0.65	0.15	-0.000975877	-9.87947E-005	0.00279992	22.0588	0.0286182
750	600	0.65	0.1	-1.65403E-005	5.49853E-005	0.000162494	22.0473	0.0136086
800	600	0.75	0.1	-7.59959E-006	3.26335E-005	9.48013E-005	22.0245	0.0203671
800	600	0.75	0.07	0.000000447	-0.000000447	1.78815E-006	22.0107	0.0189364
1000	1000	0.75	0.1	1.11759E-005	-0.000000447	3.16387E-005	22.0596	0.0134636
1000	1200	0.75	0.1	7.59959E-006	-0.000000447	2.15336E-005	22.1356	0.00927948
1200	1000	0.75	0.1	-0.000000447	-2.01166E-005	5.69231E-005	22.0172	0.0149509

Tabla 19: Resultados para la función de Ackley utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.000000447	0.000000447	1.78815E-006	21.9894	0.0199722
19	0.000000447	1.02818E-005	2.91116E-005	22.0502	0.0267565
43	-0.000000447	4.91738E-006	1.39665E-005	22.0853	0.0163699
83	1.65403E-005	-4.02331E-006	4.81548E-005	22.0983	0.00448447
101	0.000000447	1.3411E-006	3.99845E-006	22.1012	0.00975709
137	-2.23517E-006	-0.000000447	6.44737E-006	22.0213	0.0242786
151	0.000000447	-0.000000447	1.78815E-006	22.0271	0.016591
191	1.83284E-005	-0.000000447	0.000051865	22.0561	0.0263304
211	1.3411E-006	1.3411E-006	5.36451E-006	22.1084	0.0246032
233	-0.000000447	-1.47521E-005	4.17503E-005	22.0645	0.0250701
277	1.3411E-006	-5.81145E-006	1.68702E-005	22.0759	0.0252072
313	-0.000000447	-0.000000447	1.78815E-006	22.0505	0.0109177
349	0.000000447	0.000000447	1.78815E-006	21.9279	0.024105
467	1.3411E-006	2.81632E-005	0.000079769	22.0244	0.0214043
521	4.91738E-006	0.000012964	3.92221E-005	22.0716	0.0174036
599	1.3411E-006	-0.000000447	3.99845E-006	22.027	0.0170706
617	0.000126511	0.000063926	0.000401449	22.0337	0.0122426
661	-1.3411E-006	-1.20699E-005	0.000034353	22.0292	0.0245212
683	7.59959E-006	-2.45869E-005	7.28061E-005	22.0144	0.0156514
701	1.3411E-006	-2.23517E-006	7.37287E-006	22.1724	0.00388189
727	7.59959E-006	6.70552E-006	2.86688E-005	22.0512	0.0128873
743	1.47521E-005	0.000000447	4.17503E-005	22.1906	0.0254846
797	0.000000447	-0.000000447	1.78815E-006	22.0291	0.0261475
811	0.000000447	-5.81145E-006	1.64867E-005	22.0919	0.0140111
853	2.23517E-006	0.000000447	6.44737E-006	22.0178	0.00990518
883	-0.000000447	-8.49366E-006	2.40589E-005	22.0225	0.00825755
907	-1.3411E-006	8.49366E-006	2.43233E-005	22.0543	0.0142711
991	0.000000447	4.02331E-006	1.14501E-005	22.058	0.00639907

Tabla 20: Resultados para la función de Ackley utilizando Selective Crossover y una semilla de aleatorios variable.

Mejor resultado obtenido	1.78815E-006
Media de los resultados	3.637948E-005
Desviación estándar de los resultados	7.476339E-005

**B.1.5. Esfera & Cruza de 2 puntos**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.05017	-0.0366031	0.00385681	24.3723	0.000192841
50	50	0.8	0.1	-0.000245667	-0.00185761	3.51105E-006	23.8477	0.1864
100	50	0.8	0.1	0.00654694	-0.0274258	0.000795039	30.9334	0.127532
150	50	0.8	0.1	-0.00290985	-0.00256378	1.50402E-005	30.9411	0.0063052
150	150	0.8	0.1	0.000532532	-0.00140167	2.24828E-006	33.1935	0.0125533
150	150	0.7	0.15	-0.0226865	-0.00157135	0.000517145	30.9979	0.021565
150	300	0.7	0.15	-0.00664215	0.00649262	8.62722E-005	44.763	0.0257937
250	300	0.7	0.15	-0.000949402	-0.000678406	1.3616E-006	41.8795	0.03087
350	300	0.7	0.15	0.000472717	0.00117096	1.59461E-006	44.366	0.0215058
400	300	0.7	0.15	-0.00109222	-0.0023587	6.75644E-006	47.3215	0.00875548
450	300	0.7	0.15	0.000378113	-0.000671692	5.94139E-007	47.489	0.0453329
500	300	0.7	0.15	0.00089325	-0.00021759	8.4524E-007	46.8681	0.0196899
500	400	0.7	0.15	0.00089325	-0.00021759	8.4524E-007	45.574	0.0232716
500	400	0.6	0.15	0.000218201	5.79834E-006	4.76452E-008	43.8364	0.0357788
500	400	0.5	0.15	0.00155121	0.000250549	0.000002469	46.1833	0.00158113
500	400	0.5	0.18	0.000465393	0.00225311	5.29311E-006	45.05	0.0207164
500	500	0.5	0.18	0.000465393	0.00225311	5.29311E-006	46.1343	0.00943479
600	500	0.6	0.18	-0.00694916	-0.00115326	4.96208E-005	44.6623	0.00390236
600	500	0.6	0.2	0.00765717	0.00334259	6.98051E-005	47.8498	0.000845169
700	500	0.6	0.15	-0.000525208	0.000273743	3.50778E-007	46.9405	0.0117832
700	500	0.65	0.15	-4.24194E-005	8.81958E-005	9.57791E-009	48.3488	0.0389623
750	600	0.65	0.15	0.0018808	-7.84302E-005	3.54355E-006	41.5811	0.00162135
750	600	0.65	0.1	-0.000128479	-4.57764E-006	1.65278E-008	43.9684	0.00460836
800	600	0.75	0.1	9.15527E-007	0.000007019	5.01052E-011	40.835	0.0103032
800	600	0.75	0.07	-1.25122E-005	-3.05176E-007	1.56648E-010	37.9361	0.0011267
1000	1000	0.75	0.1	-7.90405E-005	-2.10571E-005	6.69081E-009	43.6108	0.000390717
1000	1200	0.75	0.1	-7.84302E-005	-2.04468E-005	6.56936E-009	43.1047	0.0198085
1200	1000	0.75	0.1	-1.52588E-006	5.95093E-005	3.54368E-009	47.9613	0.00371133

Tabla 21: Resultados para la función de Esfera utilizando Cruza de 2 puntos y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	5.88989E-005	5.88989E-005	6.93817E-009	48.5897	0.027781
19	-8.08716E-005	1.12915E-005	6.66771E-009	44.851	0.00242016
43	2.16675E-005	0.000180359	0.000000033	43.5148	0.0214962
83	1.52588E-006	-0.000005188	2.92435E-011	46.5473	0.00119451
101	6.68335E-005	5.40161E-005	7.38446E-009	43.0613	0.000867576
137	1.52588E-006	-1.80054E-005	3.26522E-010	49.4834	0.0063933
151	5.82886E-005	1.80054E-005	3.72175E-009	41.2495	0.00972436
191	-0.00250153	-9.61304E-005	6.26687E-006	42.6915	0.0214812
211	0.00018219	2.10571E-005	3.36366E-008	45.0203	0.0274487
233	-2.96021E-005	-7.53784E-005	6.55819E-009	44.7515	0.0202328
277	9.00269E-005	-4.18091E-005	9.85284E-009	45.6125	0.0113994
313	-8.23975E-006	0.000144959	2.10809E-008	42.2888	0.000251888
349	-0.000176086	-8.94165E-005	0.000000039	43.7526	0.00105768
467	-0.000122375	3.35693E-006	0.000000015	41.9002	0.0116043
521	2.22778E-005	1.06812E-005	6.10389E-010	38.1355	0.0199691
599	4.48608E-005	-0.000048523	4.36697E-009	47.5372	0.00471707
617	1.06812E-005	0.000186462	3.48823E-008	40.4374	0.0221714
661	2.16675E-005	7.90405E-005	6.71689E-009	47.5848	0.014383
683	-2.16675E-005	1.98364E-005	8.62964E-010	46.3157	0.00349983
701	-2.22778E-005	-6.98853E-005	5.38025E-009	43.4152	0.00629198
727	0.000083313	-0.00013092	2.40812E-008	36.311	0.00102566
743	0.000015564	0.000102234	1.0694E-008	45.7973	0.0102293
797	3.05176E-007	4.57764E-006	2.10479E-011	48.9406	0.00021574
811	2.74658E-006	-6.40869E-006	4.8615E-011	41.7746	0.0102804
853	-5.40161E-005	6.31714E-005	6.90837E-009	44.7222	0.0284134
883	-1.61743E-005	-0.000156555	2.47711E-008	45.5459	0.00208481
907	5.79834E-006	2.10571E-005	4.77023E-010	42.9359	0.0138534
991	8.8501E-006	-0.00007019	1.27591E-010	43.9429	0.0178827

Tabla 22: Resultados para la función Esfera utilizando Cruza de 2 puntos y una semilla de aleatorios variable.

Mejor resultado obtenido 2.10479E-011  
 Media de los resultados 2.346429E-007  
 Desviación estándar de los resultados 1.182268E-006

**B.1.6. Esfera & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.000533142	-0.00328827	0.000011097	24.0307	0.108174
50	50	0.8	0.1	0.00785126	0.000900574	6.24533E-005	26.982	0.0680412
100	50	0.8	0.1	0.000851135	-0.00139313	2.66524E-006	28.4255	0.0222802
150	50	0.8	0.1	-0.00173065	0.00291962	1.15193E-005	43.5796	0.0275796
150	150	0.8	0.1	0.00052887	0.00267792	0.000007451	33.8101	0.120246
150	150	0.7	0.15	-0.00126923	-0.000821228	2.28535E-006	34.9164	0.0368581
150	300	0.7	0.15	0.00107574	-0.000203552	1.19866E-006	32.9348	0.0886946
250	300	0.7	0.15	-0.00144073	-0.00148224	4.27275E-006	36.9419	0.0783665
350	300	0.7	0.15	-0.000395813	0.000821838	8.32086E-007	40.3469	0.0728333
400	300	0.7	0.15	-0.00108795	-0.000244446	1.24339E-006	44.5355	0.00543202
450	300	0.7	0.15	0.000745544	0.000184021	5.897E-007	44.4681	0.0133463
500	300	0.7	0.15	0.000116882	-3.02124E-005	1.45743E-008	50.2374	0.03821
500	400	0.7	0.15	0.000116882	-3.02124E-005	1.45743E-008	45.5769	0.0432895
500	400	0.6	0.15	0.00107086	7.53784E-005	1.15243E-006	41.689	0.0300343
500	400	0.5	0.15	-0.000538635	-0.00107147	1.43818E-006	41.401	0.0243302
500	400	0.5	0.18	-0.00400177	-0.0016983	1.88984E-005	48.1761	0.0284858
500	500	0.5	0.18	0.00375214	-0.00166107	1.68377E-005	44.1039	0.0453179
600	500	0.6	0.18	-0.00281647	0.00283905	1.59927E-005	47.3967	0.0154783
600	500	0.6	0.2	-0.001716	-0.00254181	9.40546E-006	46.5401	0.00295119
700	500	0.6	0.15	-5.95093E-005	5.79834E-006	3.57498E-009	45.5432	0.000961396
700	500	0.65	0.15	-0.000471497	0.000804749	8.69929E-007	48.6306	0.00843606
750	600	0.65	0.15	0.000105286	0.00214813	4.62556E-006	47.8324	0.0161714
750	600	0.65	0.1	7.96509E-005	0.000293274	9.23539E-008	45.8592	0.0092246
800	600	0.75	0.1	-6.13403E-005	-0.000104675	1.47196E-008	48.4021	0.00192834
800	600	0.75	0.07	-9.46045E-006	-2.22778E-005	5.85802E-010	38.5675	0.00784992
1000	1000	0.75	0.1	3.35693E-006	0.000015564	2.53506E-010	46.8888	0.00117325
1000	1200	0.75	0.1	1.52588E-006	3.96729E-006	1.80677E-011	50.0845	0.00163347
1200	1000	0.75	0.1	3.35693E-006	0.000179138	3.21018E-008	46.0841	0.00939738

Tabla 23: Resultados para la función de Esfera utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1200), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-8.23975E-006	2.10571E-005	5.11296E-010	41.6564	0.00292894
19	1.52588E-006	-9.15527E-007	3.1665E-012	42.2819	0.0221746
43	3.05176E-007	6.40869E-006	4.11645E-011	43.3558	0.00625777
83	0.000072937	0.000137024	2.40954E-008	45.3464	0.000714059
101	3.35693E-006	-1.52588E-006	1.35973E-011	42.785	0.0231731
137	3.05176E-007	3.96729E-006	1.58325E-011	45.245	0.0101874
151	-9.85718E-005	1.61743E-005	0.00000001	47.1825	0.0203901
191	1.52588E-006	0.000039978	1.60057E-009	42.9545	0.010356
211	3.05176E-007	-3.93677E-005	1.54991E-009	42.1645	0.0111882
233	-1.52588E-006	2.74658E-006	9.87202E-012	48.2334	0.000687268
277	-9.15527E-007	0.000005188	2.77534E-011	46.9074	0.00516961
313	-9.15527E-007	-1.00708E-005	1.02259E-010	43.5524	0.00197245
349	-3.02124E-005	-2.13623E-006	9.17353E-010	44.8717	0.00129741
467	1.06812E-005	-1.00708E-005	2.15508E-010	47.1288	0.0186236
521	-1.43433E-005	1.67847E-005	4.87454E-010	45.3876	0.0169094
599	3.05176E-007	2.22778E-005	4.96395E-010	43.896	0.0111807
617	6.40869E-006	5.79834E-006	7.46921E-011	44.5122	0.01008
661	2.74658E-006	2.13623E-006	1.21072E-011	46.5124	0.0163233
683	-9.15527E-007	3.05176E-007	9.31323E-013	47.598	0.000981924
701	9.15527E-007	-2.13623E-006	5.40167E-012	44.9783	0.0101612
727	3.05176E-007	-1.00708E-005	1.01514E-010	41.2293	0.0169652
743	-4.66919E-005	-3.50952E-005	3.41181E-009	49.3658	0.0324549
797	-1.06812E-005	1.25122E-005	2.70642E-010	47.9137	0.00110023
811	-1.98364E-005	-2.13623E-006	3.98047E-010	42.7512	0.00112402
853	-2.65503E-005	-3.05176E-007	7.05011E-010	44.4967	0.00903052
883	-1.00708E-005	-7.62939E-006	1.59629E-010	44.8207	0.00276703
907	2.13623E-006	-1.52588E-006	6.89179E-012	47.3804	0.0244132
991	3.05176E-007	-8.8501E-006	7.84174E-011	44.5151	0.0061282

Tabla 24: Resultados para la función Esfera utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	9.31323E-013
Media de los resultados	1.617522E-009
Desviación estándar de los resultados	4.814784E-009

**B.1.7. Esfera & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0463242	0.0134726	0.00232744	24.8104	0.00361572
50	50	0.8	0.1	-0.00140167	0.00317291	1.20321E-005	28.903	0.0200732
100	50	0.8	0.1	0.00714569	9.67407E-005	5.10703E-005	28.6351	0.0150905
150	50	0.8	0.1	-0.0133237	0.015834	0.000428237	34.3304	0.0469958
150	150	0.8	0.1	-0.00160919	0.000992127	3.57381E-006	35.2092	0.128944
150	150	0.7	0.15	0.0132138	0.00523163	0.000201975	34.8731	0.0933887
150	300	0.7	0.15	-0.00241364	-0.00523529	3.32339E-005	33.7688	0.0666595
250	300	0.7	0.15	-0.000751038	0.00195404	4.38233E-006	44.0825	0.00524901
350	300	0.7	0.15	1.92261E-005	0.00087616	0.000000768	49.1518	0.0573828
400	300	0.7	0.15	-3.69263E-005	-0.00143646	2.06479E-006	41.7401	0.0262415
450	300	0.7	0.15	0.00255341	0.000314636	6.61888E-006	45.0952	0.00750261
500	300	0.7	0.15	0.0018045	0.00250824	9.5475E-006	41.046	0.0141915
500	400	0.7	0.15	0.00131622	0.00254547	8.21187E-006	45.0313	0.0225609
500	400	0.6	0.15	0.000354919	0.000109558	0.000000138	45.7043	0.00995621
500	400	0.5	0.15	-0.000398254	1.92261E-005	0.000000159	43.2749	0.0189978
500	400	0.5	0.18	-0.00513153	0.00179779	2.95647E-005	44.6762	0.0257078
500	500	0.5	0.18	0.000131531	0.00171295	2.9515E-006	50.067	0.0077896
600	500	0.6	0.18	0.000918274	0.00121796	2.32665E-006	46.825	0.0490967
600	500	0.6	0.2	-0.00198029	0.00111176	5.15753E-006	48.16	0.0409539
700	500	0.6	0.15	-0.000683289	-0.000364075	5.99434E-007	44.9225	0.0118741
700	500	0.65	0.15	-0.000342102	-0.000309143	2.12603E-007	45.7911	0.0359649
750	600	0.65	0.15	0.000693665	-0.000944519	1.37329E-006	42.9946	0.0119243
750	600	0.65	0.1	6.31714E-005	1.25122E-005	4.14718E-009	43.0821	0.0116855
800	600	0.75	0.1	0.0013327	2.10571E-005	1.77654E-006	40.3337	0.0400463
800	600	0.75	0.07	5.40161E-005	8.23975E-006	0.000000003	46.4653	0.0135164
1000	1000	0.75	0.1	1.25122E-005	-2.53296E-005	7.98144E-010	46.2422	0.0232172
1000	1200	0.75	0.1	-9.15527E-007	1.25122E-005	1.57394E-010	44.9942	0.00674654
1200	1000	0.75	0.1	-6.40869E-006	-0.000039978	1.63931E-009	44.9791	0.024647

Tabla 25: Resultados para la función de Esfera utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1200), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-1.19019E-005	-3.44849E-005	1.33086E-009	45.569	0.00439493
19	-1.92261E-005	5.09644E-005	0.000000003	45.4673	0.0072722
43	3.05176E-007	1.25122E-005	1.56648E-010	48.2244	0.0126546
83	-4.54712E-005	-6.37817E-005	6.13574E-009	43.6907	0.00196586
101	4.57764E-006	-9.15527E-007	2.1793E-011	42.7817	0.0194806
137	6.86646E-005	3.96729E-006	4.73056E-009	47.3314	0.00603598
151	-9.91821E-005	5.79834E-006	9.87072E-009	44.0492	0.0116877
191	-5.79834E-006	-3.05176E-007	3.37139E-011	48.0786	0.0104559
211	8.23975E-006	-5.52368E-005	3.119E-009	43.0276	0.0199477
233	4.57764E-006	1.19019E-005	1.62609E-010	43.1911	0.0178993
277	1.31226E-005	-2.10571E-005	6.15604E-010	44.2666	0.00713411
313	9.46045E-006	1.00708E-005	1.90921E-010	46.9751	0.0246425
349	-3.32642E-005	-0.000005188	1.13342E-009	45.0185	0.00131987
467	-2.22778E-005	-1.37329E-005	6.84895E-010	43.6309	0.000375912
521	-1.25122E-005	8.23975E-006	2.24449E-010	39.3235	0.00783526
599	0.00011322	2.16675E-005	1.32883E-008	45.4687	0.0013357
617	-4.57764E-006	-1.19019E-005	1.62609E-010	43.9684	0.0245813
661	-0.000015564	-4.97437E-005	2.71667E-009	44.5236	0.00540867
683	-3.35693E-006	0.000005188	3.81842E-011	43.6242	0.00171316
701	4.57764E-006	2.74658E-006	2.84985E-011	42.3366	0.00492294
727	2.65503E-005	1.31226E-005	8.7712E-010	44.3882	0.0103254
743	-1.52588E-006	-1.49536E-005	2.25939E-010	45.1987	0.02828
797	-4.57764E-006	2.13623E-006	2.55182E-011	50.7069	0.0123957
811	0.000121765	1.00708E-005	1.49282E-008	46.3877	0.00973271
853	2.89917E-005	3.05176E-007	8.40612E-010	42.9008	0.0262027
883	1.00708E-005	-3.35693E-006	1.1269E-010	45.1496	0.0168159
907	-1.31226E-005	-8.8501E-006	2.50526E-010	47.6138	0.0280948
991	8.8501E-006	9.15527E-007	7.91624E-011	44.9151	0.00597003

Tabla 26: Resultados para la función Esfera utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.1793E-011
Media de los resultados	2.319713E-009
Desviación estándar de los resultados	0.000000004

**B.1.8. Esfera & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0318442	-0.0509677	0.00361176	10.1793	0.000180588
50	50	0.8	0.1	0.00637848	0.00829071	0.000109421	22.9733	0.146267
100	50	0.8	0.1	0.0381979	-0.0100845	0.00156078	40.728	0.147878
150	50	0.8	0.1	0.0129697	-0.0203757	0.00058338	36.179	0.0103065
150	150	0.8	0.1	0.0105161	0.00266083	0.000117667	41.7403	0.0454433
150	150	0.7	0.15	0.000340881	0.00331757	1.11224E-005	41.3159	0.0255486
150	300	0.7	0.15	-0.000797424	-0.00127716	0.000002267	35.1631	0.000413876
250	300	0.7	0.15	-0.00187531	-0.000260315	3.58453E-006	49.3676	0.0751133
350	300	0.7	0.15	0.000184631	-0.000125427	4.98207E-008	42.3294	0.0841464
400	300	0.7	0.15	0.000238953	0.00133514	1.83971E-006	45.2423	0.0170961
450	300	0.7	0.15	-0.000137024	0.00141205	2.01266E-006	43.2834	0.011554
500	300	0.7	0.15	3.57056E-005	6.40869E-006	1.31596E-009	49.0399	0.0366233
500	400	0.7	0.15	0.000007019	-6.40869E-006	9.03383E-011	43.2984	0.0153777
500	400	0.6	0.15	-0.00152863	-0.000655823	2.7668E-006	46.5003	0.0193168
500	400	0.5	0.15	-0.000746765	-0.000613709	9.34296E-007	42.9242	0.0175189
500	400	0.5	0.18	-0.0029776	0.00283905	1.69263E-005	44.7029	0.0503112
500	500	0.5	0.18	-0.00270905	0.00255646	1.38744E-005	46.0938	0.016044
600	500	0.6	0.18	-9.12476E-005	-0.000491638	0.00000025	48.3502	0.00649241
600	500	0.6	0.2	0.00500885	0.00131989	2.68307E-005	49.2654	0.0336348
700	500	0.6	0.15	0.00137299	0.000524597	2.16029E-006	45.1057	0.0429415
700	500	0.65	0.15	0.000785828	0.000549622	9.19609E-007	46.3352	0.0311645
750	600	0.65	0.15	8.75855E-005	4.24194E-005	9.47062E-009	47.525	0.00140103
750	600	0.65	0.1	7.53784E-005	0.000093689	1.44595E-008	44.9831	0.0161055
800	600	0.75	0.1	-0.000127258	-0.000193176	5.35118E-008	39.6428	0.0132351
800	600	0.75	0.07	-3.20435E-005	-1.12915E-005	1.15428E-009	40.9387	0.0113891
1000	1000	0.75	0.1	3.14331E-005	-4.66919E-005	3.16817E-009	44.5121	0.0116548
1000	1200	0.75	0.1	-2.71606E-005	-3.38745E-005	1.88518E-009	40.0127	0.000391903
1200	1000	0.75	0.1	4.18091E-005	-2.41089E-005	2.32924E-009	44.2236	0.0119202

Tabla 27: Resultados para la función de Esfera utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (500), el número de generaciones (400), la tasa de supervivencia (0,7) y la tasa de mutación (0,15). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.00172272	0.000579529	3.30361E-006	47.0788	0.0508503
19	-0.000207214	0.000100403	0.000000053	38.9208	0.00325293
43	0.00056366	-0.000176086	3.48719E-007	46.6468	0.00596478
83	0.00273651	3.87573E-005	0.00000749	44.7425	0.0190167
101	-0.000129089	0.000441589	2.11665E-007	48.9414	0.0379393
137	0.000649719	0.000679627	0.000000884	40.5883	0.00303305
151	-0.00212433	-0.000186462	4.54754E-006	50.2146	0.0501369
191	0.000071106	-0.00228668	0.000005234	38.8538	0.010279
211	0.000376282	-0.000202942	1.82773E-007	46.5975	0.0181532
233	-0.00399506	0.000662537	1.63994E-005	46.9255	0.00301702
277	-0.000366516	8.94165E-005	1.42329E-007	47.3559	0.0139748
313	0.00139679	0.00203461	6.09065E-006	47.6863	0.0331055
349	0.000514221	-0.000193787	0.000000302	42.7533	0.0346205
467	0.000851135	-4.11987E-005	7.26129E-007	41.2449	0.00483627
521	0.00649628	0.000423889	4.23813E-005	44.3699	0.0165442
599	-0.000605774	0.000395203	5.23147E-007	47.7504	0.0683303
617	-7.72095E-005	0.00101715	1.04056E-006	46.5491	0.0257291
661	-4.91333E-005	0.000707092	5.02394E-007	43.8839	0.00992189
683	-0.00175812	0.00105743	4.20915E-006	46.7994	0.0115567
701	0.00564179	0.00384979	4.66506E-005	43.2051	0.0329068
727	0.00188751	-1.52588E-006	3.56271E-006	43.2242	0.0347308
743	-0.00125153	-0.00115936	2.91044E-006	42.4161	0.00206212
797	-0.000298157	-0.0001297	1.05719E-007	44.4626	0.0334635
811	-0.00130585	0.00113129	2.98505E-006	49.7545	0.00269523
853	0.00310516	0.00182831	1.29848E-005	46.3168	0.0526698
883	0.00165924	0.000039978	2.75468E-006	45.6864	0.0256168
907	-0.000118713	0.0014743	2.18767E-006	41.6266	0.0677347
991	0.000235291	-0.000446472	2.54699E-007	42.3412	0.00396619

Tabla 28: Resultados para la función Esfera utilizando Selective Crossover y una semilla de aleatorios variable.

Mejor resultado obtenido 0.000000053  
 Media de los resultados 6.034597E-006  
 Desviación estándar de los resultados 1.155662E-005

**B.1.9. Griewank & Cruza de 2 puntos**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	2.56383	4.06911	0.197182	85.5506	3.24816
50	50	0.8	0.1	2.94622	-22.3139	0.148096	91.5835	0.77379
100	50	0.8	0.1	9.47765	-4.04602	0.0670088	108.511	0.0614705
150	50	0.8	0.1	-6.38361	9.23773	0.0672141	112.947	0.466131
150	150	0.8	0.1	6.21362	-8.8549	0.0319107	136.95	0.13854
150	150	0.7	0.15	-12.5768	8.96449	0.0612374	130.915	0.0508599
150	300	0.7	0.15	12.548	8.88997	0.0592943	140.318	0.605343
250	300	0.7	0.15	3.08805	4.57792	0.0136056	124.04	0.133807
350	300	0.7	0.15	9.39769	4.47275	0.0276701	153.705	0.016392
400	300	0.7	0.15	12.5612	-0.058673	0.0403206	155.665	0.0764561
450	300	0.7	0.15	-0.0188562	0.290436	0.0212095	169.28	0.252517
500	300	0.7	0.15	9.38153	4.53756	0.0303236	158.251	0.0252689
500	400	0.7	0.15	-9.40651	4.4378	0.0272175	152.006	0.198627
500	400	0.6	0.15	6.2251	0.0638396	0.0123924	153.757	0.301626
500	400	0.5	0.15	6.26324	0.0302835	0.0102354	144.977	0.177168
500	400	0.5	0.18	0.00838106	-8.94428	0.0208911	165.85	0.116524
500	500	0.5	0.18	0.00838106	-8.94428	0.0208911	157.521	0.0722789
600	500	0.6	0.18	-3.14878	4.59513	0.0135724	152.886	0.0879549
600	500	0.6	0.2	-3.22646	-4.50011	0.0120798	160.841	0.100032
700	500	0.6	0.15	0.0243313	-0.0747495	0.00169367	168.604	0.0920511
700	500	0.65	0.15	3.15313	13.2388	0.0483862	147.829	0.093263
750	600	0.65	0.15	3.13138	-4.57326	0.0119786	136.584	0.169154
750	600	0.65	0.1	6.27784	-0.00505513	0.0098735	123.466	0.0693444
800	600	0.75	0.1	3.13766	4.43586	0.00740051	154.529	0.0550078
800	600	0.75	0.07	-3.14026	-4.43826	0.00739608	153.137	0.00191885
1000	1000	0.75	0.1	3.14005	4.44088	0.00739753	158.123	0.0735975
1000	1200	0.75	0.1	3.14004	4.43973	0.00739646	145.438	0.0210866
1200	1000	0.75	0.1	-3.13997	4.4393	0.00739623	163.063	0.0552239

Tabla 29: Resultados para la función de Griewank utilizando Cruza de 2 puntos y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (700), el número de generaciones (500), la tasa de supervivencia (0,6) y la tasa de mutación (0,15). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-3.14268	-4.46844	0.0076247	168.027	0.113404
19	9.41102	4.44242	0.0271703	150.867	0.0345238
43	-6.28427	-8.80618	0.0308439	156.062	0.115704
83	-3.10634	4.42171	0.00803355	157.182	0.219336
101	-0.0617503	-0.00922764	0.00192817	158.405	0.105399
137	-3.15308	-4.45158	0.00752455	160.627	0.10448
151	3.13984	-4.41672	0.00751414	170.972	0.00255157
191	-6.22913	-0.00980208	0.0111854	162.632	0.0529003
211	-3.31456	4.35761	0.0242063	171.335	0.132234
233	3.1239	-4.39583	0.00798046	163.643	0.00332486
277	-6.28468	0.0469043	0.0104259	170.52	0.024246
313	-3.2136	-4.36946	0.0112905	166.269	0.123067
349	-6.23612	0.0451905	0.0113402	154.951	0.0594369
467	-3.12467	4.43496	0.00751693	154.066	0.0490483
521	0.0635698	8.91744	0.0221513	158.77	0.0488823
599	-3.23557	-4.52359	0.0137665	157.778	0.0495626
617	3.07441	4.43761	0.00954866	163.151	0.160001
661	-3.11956	-4.42569	0.0076462	156.627	0.00795408
683	3.08881	4.46286	0.00885698	157.281	0.139006
701	-3.19376	-4.47906	0.00925299	157.514	0.0894121
727	-0.00856211	-0.0100535	6.19656E-005	152.641	0.122603
743	6.31854	8.70946	0.0373259	146.39	0.0381133
797	0.0175062	-8.8684	0.0198908	166.443	0.132776
811	-3.14809	4.47656	0.00779212	159.793	0.0496016
853	0.14662	-0.156206	0.0167695	161.087	0.026624
883	-0.120024	0.0404636	0.00760461	149.435	0.114561
907	3.13975	-4.55523	0.0108073	146.644	0.0207322
991	-0.0244654	0.0221034	0.000421635	156.151	0.0731921

Tabla 30: Resultados para la función de Griewank utilizando Cruza de 2 puntos y una semilla de aleatorios variable.

Mejor resultado obtenido	6.19656E-005
Media de los resultados	0.012374
Desviación estándar de los resultados	0.008896

**B.1.10. Griewank & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	5.98336	-27.3123	0.340692	82.8791	0.360552
50	50	0.8	0.1	6.43913	-9.45835	0.124733	87.4909	0.0969534
100	50	0.8	0.1	-12.526	-8.78901	0.0616882	97.3261	0.683465
150	50	0.8	0.1	-6.2584	8.96106	0.0315907	116.573	0.362787
150	150	0.8	0.1	-6.2614	8.9561	0.0313274	102.756	0.127882
150	150	0.7	0.15	3.53395	4.58704	0.0891665	156.434	0.184636
150	300	0.7	0.15	3.51646	4.45795	0.0775569	107.561	0.248435
250	300	0.7	0.15	0.00766413	-0.0635848	0.00104095	140.658	0.021552
350	300	0.7	0.15	-6.14067	-8.69336	0.0476055	138.095	0.154831
400	300	0.7	0.15	-3.18894	13.3321	0.0481019	146.266	0.0267343
450	300	0.7	0.15	3.12373	4.4228	0.00759009	158.615	0.0805047
500	300	0.7	0.15	6.28422	0.113454	0.0130928	167.695	0.0483575
500	400	0.7	0.15	6.28411	0.113136	0.0130744	133.52	0.111278
500	400	0.6	0.15	-6.2684	8.89629	0.0297462	168.59	0.147837
500	400	0.5	0.15	-3.12939	4.45099	0.00749198	140.519	0.143242
500	400	0.5	0.18	-0.00303956	-0.0435325	0.000478826	160.148	0.0215961
500	500	0.5	0.18	-0.00303956	-0.0435325	0.000478826	153.588	0.252493
600	500	0.6	0.18	3.25689	4.40704	0.0144652	153.454	0.138386
600	500	0.6	0.2	6.22642	8.41715	0.0833272	157.354	0.226649
700	500	0.6	0.15	-6.30194	0.00258358	0.0101061	160.053	0.0800356
700	500	0.65	0.15	3.12955	4.43843	0.00745095	169.707	0.0188669
750	600	0.65	0.15	3.14704	4.4098	0.00762599	163.539	0.00299524
750	600	0.65	0.1	3.14116	-4.43912	0.0073968	144.949	0.0349149
800	600	0.75	0.1	3.12155	4.44049	0.00756773	142.045	0.023221
800	600	0.75	0.07	6.27984	0.0023053	0.00986602	142.85	0.00389753
1000	1000	0.75	0.1	3.13341	-4.43959	0.00741824	144.986	0.0428542
1000	1200	0.75	0.1	-3.13753	-4.43994	0.0073997	152.818	0.0114657
1200	1000	0.75	0.1	3.13948	-4.43452	0.00740004	150.63	0.0144163

Tabla 31: Resultados para la función de Griewank utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (500), el número de generaciones (400), la tasa de supervivencia (0,5) y la tasa de mutación (0,18). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-3.14083	-4.44463	0.00740594	149.077	0.0396175
19	-3.03954	-4.41592	0.012568	157.083	0.0584882
43	-3.23644	4.43653	0.0120436	152.6	0.233409
83	-3.228	4.59203	0.0171429	163.179	0.0270874
101	3.11637	-4.44184	0.00767879	157.941	0.208549
137	-3.1371	-4.26239	0.0151457	136.336	0.0939489
151	6.34732	0.029461	0.0123448	148.681	0.244526
191	-3.22498	4.41065	0.0111968	167.049	0.0811289
211	-3.20561	4.37387	0.0105885	150.26	0.0294125
233	-6.19544	0.112496	0.0165963	171.437	0.06771
277	0.00305297	-8.91613	0.0201094	134.801	0.0980111
313	-6.33692	8.87646	0.0312019	165.516	0.0442067
349	9.31386	4.56658	0.0368445	153.328	0.0623877
467	-6.21661	-0.183884	0.0203081	148.156	0.0889571
521	0.0326059	8.66773	0.0311689	158.499	0.102458
599	-0.0429916	0.0379854	0.00128519	157.033	0.140195
617	3.23139	-4.5415	0.0142166	160.381	0.159162
661	3.27783	-4.30382	0.0213683	144.173	0.0421069
683	0.0360548	0.054367	0.00138934	159.75	0.0472256
701	-3.20488	-4.53723	0.0119365	163.463	0.174926
727	3.07491	-4.43332	0.00952289	166.224	0.2477
743	0.00490593	-0.00607604	2.12787E-005	168.941	0.0184079
797	-9.38819	4.4568	0.0277178	164.203	0.0150688
811	3.30699	-4.451	0.0213496	167.412	0.17708
853	0.0404206	8.84554	0.0207823	155.505	0.119991
883	3.16903	4.41913	0.00791016	143.557	0.224044
907	-9.47272	4.55033	0.03164	154.633	0.0544977
991	-3.09215	4.44414	0.00855048	152.717	0.170004

Tabla 32: Resultados para la función de Griewank utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.12787E-005
Media de los resultados	0.015715
Desviación estándar de los resultados	0.009617

**B.1.11. Griewank & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-21.4848	-49.2454	0.877523	60.4678	0.164005
50	50	0.8	0.1	-2.85355	13.0582	0.103336	84.6701	0.803525
100	50	0.8	0.1	6.64949	9.12237	0.111239	106.674	0.064354
150	50	0.8	0.1	6.71058	0.19366	0.109737	102.9	0.514585
150	150	0.8	0.1	-6.25744	8.90259	0.0300051	120.429	0.0810356
150	150	0.7	0.15	-6.27437	-0.128395	0.0140032	114.136	0.140502
150	300	0.7	0.15	-6.27437	-0.128395	0.0140032	150.063	0.496013
250	300	0.7	0.15	0.0120898	-8.77719	0.0222783	125.027	0.0457755
350	300	0.7	0.15	6.25385	-0.147458	0.0156419	136.959	4.46913E-005
400	300	0.7	0.15	3.11271	13.3825	0.0483353	161.705	0.060381
450	300	0.7	0.15	0.172035	-8.89643	0.0345837	151.147	0.00951187
500	300	0.7	0.15	-3.14183	4.37837	0.00830077	158.394	0.0613505
500	400	0.7	0.15	3.14151	4.37836	0.00830034	167.126	0.218445
500	400	0.6	0.15	6.17618	0.0218485	0.0153749	149.714	0.181715
500	400	0.5	0.15	-3.11918	4.4596	0.00772544	134.96	0.115222
500	400	0.5	0.18	0.0892893	-0.0325321	0.00424942	167.725	0.254938
500	500	0.5	0.18	-0.0202878	-0.0349858	0.000512123	159.318	0.0347088
600	500	0.6	0.18	-0.0257366	8.86914	0.0200658	155.916	0.123495
600	500	0.6	0.2	3.16253	4.35533	0.00937706	155.808	0.0443806
700	500	0.6	0.15	-6.24579	-8.86325	0.0302177	171.025	0.0466242
700	500	0.65	0.15	3.12835	-4.44302	0.00746945	162.152	0.128596
750	600	0.65	0.15	0.0421126	-0.0220793	0.00100893	161.009	0.0919703
750	600	0.65	0.1	-6.27029	0.000577513	0.00991233	147.456	0.099845
800	600	0.75	0.1	3.14334	-4.43761	0.00740172	151.142	0.010063
800	600	0.75	0.07	3.14013	-4.43728	0.00739638	123.792	0.0191628
1000	1000	0.75	0.1	-3.14017	-4.438	0.0073961	167.109	0.0574313
1000	1200	0.75	0.1	-3.14005	-4.43802	0.00739609	159.331	0.0792526
1200	1000	0.75	0.1	0.0051518	-0.00135424	1.37361E-005	162.305	0.0419177

Tabla 33: Resultados para la función de Griewank utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1200), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.00179457	-0.000851881	1.79264E-006	144.077	0.00214038
19	-6.27745	-0.0072998	0.00988138	147.669	0.0100152
43	-3.13918	-4.43778	0.0073965	157.291	0.00672039
83	0.00442537	-0.00239192	1.12285E-005	159.238	0.102999
101	-3.13842	-4.43528	0.00739983	170.387	0.0390499
137	0.0357592	0.00567483	0.000647664	165.191	0.0973014
151	6.27749	0.000183005	0.00986793	158.418	0.0479403
191	-0.0289385	-0.0231729	0.00055322	142.624	0.0146357
211	-0.00187838	0.0373478	0.000350808	157.024	0.0855411
233	-3.13894	-4.43877	0.00739665	156.073	0.0508095
277	6.27821	-0.00705114	0.00987879	171.369	0.0125158
313	3.13986	4.44732	0.00741577	164.172	0.0112919
349	-3.12843	-4.43389	0.00746848	155.594	0.0116917
467	-3.13988	-4.43829	0.00739606	158.325	0.00604383
521	-0.00219801	0.00269031	4.22809E-006	145.672	0.0169949
599	-3.13908	-4.43831	0.00739649	159.651	0.0124958
617	-0.00356594	0.0028166	8.34642E-006	148.694	0.00274464
661	-6.27978	-0.000184122	0.00986471	155.667	0.124389
683	3.14258	-4.43632	0.00740044	168.851	0.00175863
701	6.28025	2.20723E-005	0.00986469	166.853	0.00734808
727	-0.000808295	-0.0037889	3.91936E-006	164.719	0.0789269
743	-6.27991	-0.000230502	0.00986469	154.541	0.0128709
797	-0.0229527	-0.0522581	0.000946686	155.361	0.0395132
811	0.00920026	0.000251178	0.000042359	164.929	0.0100477
853	3.15175	-4.44557	0.00747752	157.731	0.073122
883	3.14085	4.43995	0.00739695	161.935	0.0617971
907	3.14043	-4.43858	0.00739613	172.895	0.0748345
991	3.13995	-4.43859	0.00739605	161.794	0.0666797

Tabla 34: Resultados para la función de Griewank utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	1.79264E-006
Media de los resultados	0.005383
Desviación estándar de los resultados	0.004009

**B.1.12. Griewank & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-27.781	5.13252	0.421427	78.5223	1.3097
50	50	0.8	0.1	-3.37391	13.2421	0.0753723	103.581	0.0676286
100	50	0.8	0.1	3.24924	13.3447	0.0530114	95.1099	0.132062
150	50	0.8	0.1	0.200357	-8.85783	0.039821	92.6612	0.113101
150	150	0.8	0.1	0.0554544	0.0141121	0.00158773	113.39	0.0626879
150	150	0.7	0.15	-2.96878	4.40349	0.0223288	143.03	0.211923
150	300	0.7	0.15	3.1859	4.39244	0.00897784	118.33	0.210504
250	300	0.7	0.15	-3.17547	-4.47839	0.00842392	129.273	0.166791
350	300	0.7	0.15	0.0779693	0.0562401	0.00382861	148.419	0.0797257
400	300	0.7	0.15	-3.11526	-4.58685	0.0132082	157.027	0.0240997
450	300	0.7	0.15	-0.0579203	-0.0193865	0.00177165	158.894	0.049174
500	300	0.7	0.15	-3.12686	-4.44338	0.00748875	159.737	0.157665
500	400	0.7	0.15	-3.12686	-4.44338	0.00748875	134.573	0.054915
500	400	0.6	0.15	-6.30968	0.114368	0.0135745	164.746	0.0460841
500	400	0.5	0.15	9.46608	-4.46196	0.0283228	163.56	0.112753
500	400	0.5	0.18	-3.23702	4.35641	0.0137745	153.825	0.0206315
500	500	0.5	0.18	-3.24496	4.4418	0.0129031	145.114	0.174988
600	500	0.6	0.18	-3.17481	4.42369	0.00805586	168.498	0.0620798
600	500	0.6	0.2	3.15156	4.41216	0.00763549	165.155	0.00494747
700	500	0.6	0.15	-3.10636	-4.48133	0.0084228	155.233	0.064721
700	500	0.65	0.15	3.13554	4.41128	0.00759071	159.457	0.0418791
750	600	0.65	0.15	-6.32509	-0.0586613	0.0117399	167.872	0.0715746
750	600	0.65	0.1	-0.000655744	-0.0115606	3.36602E-005	157.976	0.0658561
800	600	0.75	0.1	-0.00167219	-1.59256E-005	1.39887E-006	159.263	0.00828707
800	600	0.75	0.07	6.29131	0.00149673	0.00992872	157.423	0.0134854
1000	1000	0.75	0.1	-3.14094	4.43409	0.00740121	156.68	0.0178443
1000	1200	0.75	0.1	-3.14109	4.43451	0.00740047	164.569	0.0555511
1200	1000	0.75	0.1	3.14015	4.43842	0.00739605	138.156	0.0597988

Tabla 35: Resultados para la función de Griewank utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.0947594	-0.00644932	0.00449892	150.116	0.015517
19	9.41934	4.43932	0.0271258	141.326	0.0244785
43	6.26987	-0.00176271	0.0099172	143.234	0.0138435
83	-6.29162	-0.00177668	0.00993253	159.636	0.0254274
101	3.13502	4.43822	0.0074086	172.002	0.0179495
137	9.41902	4.43829	0.0271259	162.291	0.00316221
151	-3.13536	-4.43974	0.00740734	163.859	0.0105396
191	6.27977	-0.00391966	0.00986855	157.571	0.047924
211	-3.13879	4.43687	0.00739742	153.29	0.0415031
233	0.000298116	-0.000153948	5.03898E-008	143.045	0.0235928
277	6.2779	-0.00975905	0.00989081	162.523	0.0217457
313	6.28175	0.00459859	0.00987141	162.85	0.0251085
349	-6.28498	0.000184122	0.00987684	146.095	0.109822
467	3.14048	4.4386	0.00739615	150.265	0.0732866
521	3.139	4.43859	0.00739657	153.323	0.00264558
599	3.13797	4.44493	0.00740867	146.967	0.135508
617	-6.26212	0.00139168	0.0100259	155.719	0.0048591
661	-3.13532	-4.46779	0.00762265	160.801	0.00287515
683	3.14084	4.44031	0.00739725	143.49	0.100648
701	3.16343	4.45751	0.00776111	154.094	0.142862
727	6.27427	-0.000554044	0.00988141	141.861	0.0926801
743	3.1331	-4.47809	0.00781326	147.93	0.0212625
797	-3.14013	4.48086	0.0078463	157.79	0.0654587
811	-3.14031	-4.43866	0.00739609	144.324	0.126871
853	3.14636	-4.42964	0.00743553	150.324	0.0931257
883	-3.13768	4.43891	0.00739884	153.97	0.0534175
907	-3.13946	4.43838	0.0073962	147.357	0.0807938
991	3.13807	4.43334	0.00740446	158.174	0.0500402

Tabla 36: Resultados para la función de Griewank utilizando Selective Crossover y una semilla de aleatorios variable.

Mejor resultado obtenido	5.03898E-008
Media de los resultados	0.009211
Desviación estándar de los resultados	0.005445

**B.1.13. Rastringin & Cruza de 2 puntos**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.0681039	-0.00412689	0.90967	34.4158	1.30424
50	50	0.8	0.1	-0.0186783	-0.985973	1.08008	47.4911	0.646839
100	50	0.8	0.1	0.00030426	0.00165497	0.000561744	59.9694	0.422512
150	50	0.8	0.1	0.0151413	0.000570984	0.0455139	55.6617	0.0178773
150	150	0.8	0.1	-0.000124207	-0.000457459	0.000044578	63.3893	0.0764294
150	150	0.7	0.15	-0.000727844	0.00127655	0.000428396	62.0223	0.364674
150	300	0.7	0.15	0.000522766	-0.00127228	0.000375353	56.2806	0.120211
250	300	0.7	0.15	-0.000516663	-0.000674744	0.000143283	72.2443	0.102619
350	300	0.7	0.15	0.00409393	-0.000204773	0.00333326	61.2038	0.0451422
400	300	0.7	0.15	5.52368E-005	-0.000652771	8.51425E-005	76.4502	0.0498079
450	300	0.7	0.15	-0.000470276	6.92749E-005	4.48285E-005	69.3416	0.113816
500	300	0.7	0.15	0.000434876	0.000317078	5.74655E-005	61.0365	0.0649665
500	400	0.7	0.15	0.000343933	0.000346985	4.73541E-005	77.7197	0.0377065
500	400	0.6	0.15	4.11987E-005	-0.000027771	4.89746E-007	67.1242	0.0710069
500	400	0.5	0.15	5.58472E-005	-0.000232849	1.13754E-005	69.2475	0.0368933
500	400	0.5	0.18	-0.00396454	-0.00264984	0.00451111	66.0809	0.0955825
500	500	0.5	0.18	-0.0027652	-0.00257294	0.00283027	66.0184	0.0960093
600	500	0.6	0.18	8.45337E-005	0.000995789	0.000198143	76.4597	0.0546648
600	500	0.6	0.2	0.0018277	0.000227966	0.000673031	71.2737	0.0668071
700	500	0.6	0.15	0.000662537	0.000607605	0.000160329	70.7982	0.0141224
700	500	0.65	0.15	-8.39233E-005	0.00533356	0.00564453	71.1599	0.0605427
750	600	0.65	0.15	-5.34058E-005	0.000128479	3.8407E-006	70.9652	0.00784728
750	600	0.65	0.1	2.13623E-006	3.05176E-007	9.23839E-010	68.5339	0.0367332
800	600	0.75	0.1	-0.000085144	4.42505E-005	1.82673E-006	75.698	0.0249492
800	600	0.75	0.07	0.000005188	-2.74658E-006	6.83641E-009	69.7354	0.0145928
1000	1000	0.75	0.1	2.10571E-005	2.04468E-005	1.7091E-007	67.8458	0.0230074
1000	1200	0.75	0.1	1.80054E-005	2.04468E-005	1.4726E-007	66.7599	0.0368631
1200	1000	0.75	0.1	2.13623E-006	5.79834E-006	7.57549E-009	67.9423	0.0206163

Tabla 37: Resultados para la función de Rastringin utilizando Cruza de 2 puntos y una semilla de aleatorios fija.



**B.1.14. Rastringin & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	1.00013	0.966049	2.16014	41.7049	0.108109
50	50	0.8	0.1	-0.00378693	-0.0039267	0.00590384	48.721	0.628732
100	50	0.8	0.1	-0.00417511	-0.00476898	0.00796985	53.7372	0.336731
150	50	0.8	0.1	-0.000970764	-0.00235016	0.00128272	59.5066	0.0683985
150	150	0.8	0.1	-2.34985E-005	-0.000166931	0.000005638	58.3088	0.0908364
150	150	0.7	0.15	0.000150452	-0.00517609	0.00531935	65.4143	0.241839
150	300	0.7	0.15	0.000119934	-0.00514191	0.00524775	65.5122	0.189094
250	300	0.7	0.15	-0.000482483	-0.000062561	4.69603E-005	65.8639	0.136005
350	300	0.7	0.15	6.43921E-005	0.000178528	7.14582E-006	69.7575	0.0863073
400	300	0.7	0.15	0.000184631	4.54712E-005	7.17317E-006	65.8707	0.065293
450	300	0.7	0.15	-0.000815735	0.000641174	0.000213575	76.8202	0.0501444
500	300	0.7	0.15	3.50952E-005	3.35693E-006	2.46591E-007	62.0938	0.0813162
500	400	0.7	0.15	3.96729E-006	-2.28882E-005	1.07055E-007	66.5664	0.0808831
500	400	0.6	0.15	-0.000224304	-5.34058E-005	1.05475E-005	71.0781	0.0715469
500	400	0.5	0.15	-0.000201111	0.000107727	1.03265E-005	67.4176	0.0360745
500	400	0.5	0.18	-0.000160828	0.00223785	0.000998666	69.7601	0.0339526
500	500	0.5	0.18	-4.57764E-006	-0.000105896	2.22893E-006	70.9316	0.10957
600	500	0.6	0.18	-0.00133636	9.15527E-007	0.000354302	70.3367	0.0571837
600	500	0.6	0.2	0.000754089	-5.52368E-005	0.000113421	71.063	0.0322269
700	500	0.6	0.15	-0.00199371	-5.88989E-005	0.000789269	77.3631	0.0140878
700	500	0.65	0.15	-5.79834E-006	3.96729E-006	9.7927E-009	72.6363	0.0486004
750	600	0.65	0.15	-1.19019E-005	-0.000281677	0.000015769	69.6769	0.00855905
750	600	0.65	0.1	-0.000101013	-3.44849E-005	2.26027E-006	68.5867	0.0388568
800	600	0.75	0.1	1.52588E-006	3.96729E-006	3.5845E-009	65.8285	0.0153301
800	600	0.75	0.07	-9.15527E-007	-3.05176E-007	1.84768E-010	66.5595	0.0462136
1000	1000	0.75	0.1	6.40869E-006	2.22778E-005	1.06611E-007	62.3405	0.0371407
1000	1200	0.75	0.1	3.05176E-007	-2.10571E-005	0.000000088	72.2465	0.0281491
1200	1000	0.75	0.1	2.13623E-006	2.13623E-006	1.81073E-009	70.7542	0.0412762

Tabla 39: Resultados para la función de Rastringin utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-9.15527E-007	3.05176E-007	1.84768E-010	62.3631	0.016046
19	3.05176E-007	3.35693E-006	2.25417E-009	59.0616	0.0204107
43	2.13623E-006	4.57764E-006	5.06264E-009	60.0838	0.0531146
83	-9.15527E-007	3.05176E-007	1.84768E-010	61.0455	0.0617824
101	2.74658E-006	-2.74658E-006	0.000000003	56.7693	0.0313295
137	-3.05176E-007	9.15527E-007	1.84768E-010	63.9187	0.047632
151	3.05176E-007	9.15527E-007	1.84768E-010	61.4682	0.0393468
191	3.05176E-007	-3.05176E-007	3.69518E-011	65.8467	0.037378
211	-9.15527E-007	-3.35693E-006	2.40198E-009	70.1285	0.040236
233	-3.05176E-007	-3.05176E-007	3.69518E-011	62.5842	0.0120845
277	3.05176E-007	3.05176E-007	3.69518E-011	59.967	0.0450656
313	-1.43433E-005	-9.15527E-007	0.000000041	67.2024	0.0350105
349	-0.000027771	3.05176E-007	0.000000153	63.8536	0.0778713
467	2.74658E-006	-3.05176E-007	1.5151E-009	66.2705	0.0439528
521	-1.52588E-006	-3.05176E-007	4.80396E-010	64.9752	0.0342674
599	9.15527E-007	3.05176E-007	1.84768E-010	68.7972	0.0439975
617	-1.86157E-005	-1.61743E-005	1.20653E-007	61.3624	0.0333455
661	3.05176E-007	-1.52588E-006	4.80396E-010	64.0207	0.031142
683	2.13623E-006	3.05176E-007	9.23839E-010	62.4836	0.00983519
701	-3.05176E-007	-3.05176E-007	3.69518E-011	62.9652	0.0190097
727	9.15527E-007	3.05176E-007	1.84768E-010	62.9996	0.037142
743	-9.15527E-007	-3.05176E-007	1.84768E-010	60.3567	0.0110004
797	-9.15527E-007	-7.62939E-006	1.17143E-008	57.7405	0.0242228
811	9.15527E-007	9.15527E-007	3.32584E-010	63.7349	0.0247668
853	-1.06812E-005	-1.52588E-006	2.3096E-008	64.8486	0.0383803
883	-2.13623E-006	0.000005188	6.24516E-009	61.5022	0.0269994
907	-1.00708E-005	1.52588E-006	2.05831E-008	70.653	0.0628602
991	-7.62939E-006	2.13623E-006	1.24534E-008	62.8059	0.0433291

Tabla 40: Resultados para la función de Rastringin utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	3.69518E-011
Media de los resultados	1.452274E-008
Desviación estándar de los resultados	3.60E-008

**B.1.15. Rastringin & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0106314	-0.998577	1.01997	36.7669	0.924833
50	50	0.8	0.1	0.012753	0.00190765	0.0329712	53.4465	0.495335
100	50	0.8	0.1	-0.0110757	0.0160703	0.0755198	51.3471	0.302625
150	50	0.8	0.1	-0.00335968	0.0150815	0.0473304	66.2496	0.104741
150	150	0.8	0.1	0.000309753	-0.000474548	6.37126E-005	57.9346	0.260172
150	150	0.7	0.15	-0.000106506	-0.00016449	7.61839E-006	71.8287	0.230087
150	300	0.7	0.15	-0.000106506	-0.00016449	7.61839E-006	58.7062	0.258486
250	300	0.7	0.15	-2.89917E-005	-0.00131805	0.000344826	67.4732	0.0780618
350	300	0.7	0.15	0.000299988	-1.37329E-005	1.78913E-005	69.7954	0.133394
400	300	0.7	0.15	-0.0014328	4.18091E-005	0.000407628	70.6817	0.0296727
450	300	0.7	0.15	-0.00234589	-0.00110992	0.00133618	70.6294	0.0522346
500	300	0.7	0.15	-0.000704651	-0.000135803	0.000102167	66.3535	0.0753547
500	400	0.7	0.15	0.00054657	-0.000118713	6.20635E-005	66.6417	0.0789458
500	400	0.6	0.15	-0.000245667	-0.000191956	1.92836E-005	68.9901	0.0480319
500	400	0.5	0.15	4.57764E-006	-3.93677E-005	3.1163E-007	67.583	0.0377214
500	400	0.5	0.18	-0.000833435	0.000195618	0.000145398	72.5715	0.0687752
500	500	0.5	0.18	-0.000833435	0.000195618	0.000145398	77.9962	0.0717496
600	500	0.6	0.18	0.0018222	0.00201813	0.00146675	69.9848	0.0455339
600	500	0.6	0.2	0.00185761	0.00107697	0.000914693	80.148	0.0557499
700	500	0.6	0.15	-0.000317078	0.000171814	2.58026E-005	70.1427	0.029652
700	500	0.65	0.15	-9.49097E-005	0.000700989	9.92743E-005	78.0711	0.0131335
750	600	0.65	0.15	0.000155334	6.01196E-005	5.50405E-006	69.4909	0.0467041
750	600	0.65	0.1	-0.000318909	-0.000178528	2.65003E-005	61.312	0.0483499
800	600	0.75	0.1	-3.08228E-005	-1.37329E-005	2.25897E-007	68.4395	0.0337663
800	600	0.75	0.07	-7.62939E-006	-0.000007019	2.13222E-008	62.8074	0.00871773
1000	1000	0.75	0.1	5.79834E-006	-3.57056E-005	2.59599E-007	69.6354	0.0424256
1000	1200	0.75	0.1	5.79834E-006	3.26538E-005	2.18211E-007	74.2272	0.039865
1200	1000	0.75	0.1	-3.14331E-005	1.12915E-005	2.21315E-007	67.1737	0.016766

Tabla 41: Resultados para la función de Rastringin utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	2.13623E-006	0.000017395	6.09365E-008	59.9543	0.0390861
19	9.15527E-007	-6.19507E-005	7.61576E-007	62.7937	0.00765323
43	0.000005188	0.000007019	1.5114E-008	59.7678	0.0186027
83	4.05884E-005	2.10571E-005	4.14804E-007	65.5615	0.0453913
101	-4.11987E-005	-2.04468E-005	4.19682E-007	66.4977	0.0354663
137	3.05176E-007	-7.62939E-006	1.15665E-008	61.1969	0.0215695
151	1.98364E-005	6.40869E-006	8.62127E-008	69.7602	0.033444
191	-9.15527E-007	3.05176E-007	1.84768E-010	64.0373	0.0149455
211	1.52588E-006	2.13623E-006	1.36728E-009	63.173	0.0513258
233	3.05176E-007	7.62939E-006	1.15665E-008	61.7951	0.0317404
277	1.37329E-005	1.52588E-006	3.78774E-008	64.6679	0.0446091
313	1.06812E-005	-0.000017395	8.26652E-008	63.7757	0.0250135
349	3.96729E-006	0.000015564	5.11807E-008	66.0319	0.036347
467	3.05176E-007	-5.79834E-006	6.6886E-009	63.5032	0.0211414
521	-9.15527E-007	-1.52588E-006	6.28212E-010	61.5649	0.0176103
599	3.05176E-007	7.84302E-005	1.22039E-006	64.0632	0.025508
617	-2.13623E-006	3.05176E-007	9.23839E-010	59.6975	0.0123155
661	-3.05176E-007	1.52588E-006	4.80396E-010	58.7851	0.0443869
683	2.74658E-006	3.35693E-006	3.73231E-009	61.0965	0.0261626
701	1.12915E-005	-9.46045E-006	4.30509E-008	65.2874	0.0222616
727	1.37329E-005	-8.8501E-006	0.000000053	70.6633	0.042855
743	-9.15527E-007	7.62939E-006	1.17143E-008	67.3371	0.0326575
797	3.05176E-007	1.52588E-006	4.80396E-010	67.1916	0.0507522
811	-2.13623E-006	-4.18091E-005	3.47696E-007	68.5609	0.0306279
853	-0.000005188	2.74658E-006	6.83641E-009	67.9731	0.023521
883	-3.35693E-006	1.25122E-005	3.32952E-008	69.4773	0.0131182
907	-1.52588E-006	3.35693E-006	2.69761E-009	64.5593	0.020232
991	0.000083313	8.8501E-006	1.3926E-006	69.5868	0.0265188

Tabla 42: Resultados para la función de Rastrigin utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	1.84768E-010
Media de los resultados	1.813894E-007
Desviación estándar de los resultados	3.643900E-007

**B.1.16. Rastrigin & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.0107663	-0.992776	1.01888	37.7647	0.803735
50	50	0.8	0.1	0.00398163	0.0113327	0.028614	48.0328	0.960656
100	50	0.8	0.1	-0.00875519	-0.00286774	0.0168352	53.5057	0.000168352
150	50	0.8	0.1	0.0106851	0.0031131	0.0245651	62.0876	0.214967
150	150	0.8	0.1	-0.000626526	0.00226898	0.00109924	56.8791	0.264565
150	150	0.7	0.15	-0.000206604	0.00280487	0.00156925	66.1157	0.233541
150	300	0.7	0.15	-0.000126648	-0.000329285	2.46936E-005	60.5638	0.340787
250	300	0.7	0.15	-0.00506561	0.00391388	0.00812935	65.4109	0.085642
350	300	0.7	0.15	-0.00267731	-0.00115326	0.00168591	71.3938	0.160437
400	300	0.7	0.15	0.00193268	-0.00111176	0.00098625	70.4303	0.0467019
450	300	0.7	0.15	-0.00250397	0.000215149	0.00125305	72.1404	0.113651
500	300	0.7	0.15	3.75366E-005	3.50952E-005	5.23891E-007	65.7062	0.0395145
500	400	0.7	0.15	0.000038147	3.32642E-005	5.08223E-007	65.0034	0.0624501
500	400	0.6	0.15	-0.000848084	-0.00239899	0.00128445	67.7795	0.119457
500	400	0.5	0.15	0.000353088	-3.57056E-005	2.49869E-005	70.4547	0.0539288
500	400	0.5	0.18	-0.000826721	0.00290924	0.00181468	70.2706	0.0967809
500	500	0.5	0.18	-0.000826721	0.00290924	0.00181468	76.127	0.0519912
600	500	0.6	0.18	0.000135803	-0.000649109	8.72501E-005	77.9842	0.0560179
600	500	0.6	0.2	-0.000573425	-0.00242462	0.00123152	76.9839	0.0620382
700	500	0.6	0.15	0.00108063	0.00107391	0.000460478	65.9478	0.044054
700	500	0.65	0.15	0.000202332	-0.00110199	0.000249046	73.3852	0.0650506
750	600	0.65	0.15	-0.000283508	0.000632019	9.51938E-005	75.0996	0.0413822
750	600	0.65	0.1	2.10571E-005	-4.54712E-005	4.98171E-007	67.0099	0.0445756
800	600	0.75	0.1	3.08228E-005	3.63159E-005	4.50132E-007	67.1993	0.0652896
800	600	0.75	0.07	-9.15527E-007	-3.05176E-007	1.84768E-010	66.0755	0.0302986
1000	1000	0.75	0.1	9.15527E-007	3.05176E-007	1.84768E-010	72.5698	0.0227691
1000	1200	0.75	0.1	-9.15527E-007	3.05176E-007	1.84768E-010	72.9758	0.0163646
1200	1000	0.75	0.1	3.05176E-007	1.52588E-006	4.80396E-010	71.9544	0.0176323

Tabla 43: Resultados para la función de Rastrigin utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1200), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-2.74658E-006	3.05176E-007	1.5151E-009	68.0852	0.0302285
19	-1.52588E-006	-3.35693E-006	2.69761E-009	67.0729	0.0362767
43	-9.15527E-007	-3.05176E-007	1.84768E-010	68.6163	0.0220909
83	5.79834E-006	9.15527E-007	6.83641E-009	67.3018	0.0384419
101	-0.000007019	-5.79834E-006	1.64443E-008	68.1171	0.00968955
137	-1.19019E-005	1.00708E-005	4.82244E-008	65.7212	0.0399441
151	-1.00708E-005	9.15527E-007	2.02875E-008	65.9575	0.0372549
191	-2.13623E-006	-3.05176E-007	9.23839E-010	68.2109	0.0424643
211	4.57764E-006	-9.15527E-007	4.32357E-009	76.3707	0.0107033
233	-3.05176E-007	3.05176E-007	3.69518E-011	68.3291	0.0255583
277	-7.62939E-006	-4.42505E-005	0.0000004	68.8899	0.00192473
313	3.05176E-007	-7.62939E-006	1.15665E-008	69.9668	0.0357985
349	3.96729E-006	3.35693E-006	5.35827E-009	75.682	0.0293222
467	-1.12915E-005	5.79834E-006	0.000000032	66.4014	0.00768791
521	-3.05176E-007	-9.15527E-007	1.84768E-010	66.2415	0.0418207
599	-5.79834E-006	-9.15527E-007	6.83641E-009	69.1736	0.0206296
617	-3.35693E-006	-6.40869E-006	1.0384E-008	71.9559	0.0205272
661	-3.05176E-007	1.43433E-005	4.08337E-008	67.7507	0.032379
683	-1.52588E-006	-2.59399E-005	0.000000134	72.1749	0.0202103
701	6.40869E-006	-9.15527E-007	8.31456E-009	72.1309	0.0367864
727	1.52588E-006	-2.74658E-006	0.000000002	71.518	0.00489435
743	-2.74658E-006	6.40869E-006	9.64489E-009	70.0226	0.0567121
797	-3.35693E-006	5.79834E-006	8.90581E-009	75.3131	0.0250345
811	1.61743E-005	4.11987E-005	3.88641E-007	68.2792	0.0546958
853	-1.80054E-005	-3.05176E-007	6.43362E-008	67.3551	0.00604354
883	-3.05176E-007	2.74658E-006	1.5151E-009	70.8087	0.0037078
907	-1.98364E-005	-3.05176E-007	7.80829E-008	66.0578	0.0176863
991	0.000235291	2.28882E-005	1.10873E-005	68.6512	0.0108728

Tabla 44: Resultados para la función de Rastrigin utilizando Selective Crossover y una semilla de aleatorios variable.

Mejor resultado obtenido	3.69518E-011
Media de los resultados	4.425458E-007
Desviación estándar de los resultados	2.088683E-006

**B.1.17. Rosenbrock & Cruza de 2 puntos**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	2.68327	7.39095	6.4817	3845200	281.772
50	50	0.8	0.1	1.15007	1.21515	1.17836	52204800	80606
100	50	0.8	0.1	1.00357	1.00621	0.000103564	58894800	170.885
150	50	0.8	0.1	1.9954	3.985	0.991964	71008600	145054
150	150	0.8	0.1	1.29013	1.66731	0.0849982	66723800	9675.23
150	150	0.7	0.15	0.892049	0.798525	0.0124233	63237200	407.672
150	300	0.7	0.15	0.928586	0.862287	0.00510003	55714500	2069.65
250	300	0.7	0.15	1.04392	1.09505	0.00471062	75694900	59235.8
350	300	0.7	0.15	1.14857	1.3065	0.0382204	74037500	548.077
400	300	0.7	0.15	1.23562	1.52351	0.0565649	77304900	14.5235
450	300	0.7	0.15	0.990378	0.964769	0.0259516	77032000	2932.21
500	300	0.7	0.15	1.13474	1.28783	0.0181585	76585800	15745.2
500	400	0.7	0.15	1.13474	1.28783	0.0181585	79023900	8974.09
500	400	0.6	0.15	1.0137	1.02934	0.000490663	79037200	12375.8
500	400	0.5	0.15	1.04691	1.09736	0.00237847	69975500	68.0645
500	400	0.5	0.18	0.99557	0.991532	3.34294E-005	75446300	14557.7
500	500	0.5	0.18	0.99557	0.991532	3.34294E-005	81214000	21.4591
600	500	0.6	0.18	1.09278	1.19922	0.0111549	82905100	19.2366
600	500	0.6	0.2	1.0508	1.09827	0.00608984	80109100	377.528
700	500	0.6	0.15	0.957198	0.916265	0.00183218	80534500	10.3452
700	500	0.65	0.15	1.04296	1.08794	0.00184876	80074000	47955.5
750	600	0.65	0.15	1.13414	1.28642	0.017996	82446500	726.97
750	600	0.65	0.1	0.941805	0.886747	0.00339291	82137700	1824.02
800	600	0.75	0.1	1.06333	1.13068	0.00401059	82439200	59.6045
800	600	0.75	0.07	0.876353	0.76711	0.0153669	83551700	1.75865
1000	1000	0.75	0.1	0.980915	0.961986	0.000368546	75883400	0.742508
1000	1200	0.75	0.1	0.980901	0.961986	0.000368036	76937800	59878.1
1200	1000	0.75	0.1	0.992292	0.984589	5.97043E-005	80769300	209.462

Tabla 45: Resultados para la función de Rosenbrock utilizando Cruza de 2 puntos y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (500), el número de generaciones (500), la tasa de supervivencia (0,5) y la tasa de mutación (0,18). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.802255	0.640204	0.0402653	74939600	8314.08
19	0.962081	0.926856	0.00159549	83449900	10025.6
43	0.977943	0.953659	0.00122264	81075600	558.246
83	1.06906	1.14427	0.00495845	75846600	17.9542
101	1.03579	1.0723	0.00131308	81386000	24549.7
137	0.864874	0.772866	0.0800512	82575000	50.3449
151	1.04218	1.08317	0.00266493	78399600	5770.98
191	0.940532	0.878636	0.00709467	76294500	19747.7
211	0.906874	0.819971	0.00927274	80517500	2670.59
233	1.03273	1.06746	0.00115552	82489300	3213.6
277	1.00323	0.999847	0.00440926	77653400	5879.59
313	1.00129	1.00341	7.04549E-005	75823700	8754.12
349	1.06138	1.12678	0.00377397	77283100	34050.5
467	0.825614	0.678355	0.0314884	77357600	1533.29
521	1.05731	1.09595	0.0515033	78392400	1192.8
599	0.964399	0.939444	0.0100619	76110200	14077.5
617	1.03385	1.06943	0.0011799	81966000	35804.5
661	1.03526	1.07389	0.00169684	78995400	18266.2
683	1.20152	1.43852	0.0432395	81623400	1433.03
701	1.20655	1.45261	0.0436494	78478100	19989
727	1.03204	1.05037	0.0227463	79018900	9174.64
743	1.04289	1.08762	0.00183923	82340200	1573.24
797	0.943619	0.891838	0.00338096	81328000	46729.7
811	1.00738	1.01326	0.000295748	80427200	45074.8
853	0.994586	0.986412	0.00080706	81168300	8020.66
883	0.894945	0.802191	0.0111968	79503200	21748.7
907	1.02532	1.0401	0.0131153	79483000	2996.52
991	1.08042	1.16539	0.00683347	82347000	83.1963

Tabla 46: Resultados para la función de Rosenbrock utilizando Cruza de 2 puntos y una semilla de aleatorios variable.

Mejor resultado obtenido	7.04549E-005
Media de los resultados	0.014317
Desviación estándar de los resultados	0.020092

**B.1.18. Rosenbrock & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	1.88484	3.56061	0.789334	3940910	197045
50	50	0.8	0.1	1.21933	1.20097	8.21661	44659300	2001.96
100	50	0.8	0.1	1.56821	2.41418	0.526373	50506400	350.591
150	50	0.8	0.1	-0.758868	0.617841	3.26968	49048000	1017.04
150	150	0.8	0.1	0.757044	0.571148	0.0594147	74231400	10073
150	150	0.7	0.15	0.640934	0.39083	0.168796	74151500	18726.6
150	300	0.7	0.15	0.765317	0.566643	0.0914311	75831800	336.516
250	300	0.7	0.15	0.663282	0.436975	0.11426	77438400	307.44
350	300	0.7	0.15	1.09441	1.18969	0.0153727	71375100	60388.8
400	300	0.7	0.15	0.972447	0.960019	0.0213982	83145500	65854.1
450	300	0.7	0.15	0.94843	0.903124	0.00395832	75373700	561.54
500	300	0.7	0.15	0.839433	0.705708	0.0258943	79623000	574.848
500	400	0.7	0.15	0.854052	0.730412	0.0214019	77378000	3071.7
500	400	0.6	0.15	1.03466	1.06955	0.00129365	82375900	10494.5
500	400	0.5	0.15	0.861398	0.741144	0.0192849	78363900	9015.41
500	400	0.5	0.18	1.00309	1.01198	0.00336492	80149100	3859.52
500	500	0.5	0.18	0.988901	0.980284	0.000679363	75657500	0.280529
600	500	0.6	0.18	0.857327	0.728492	0.0246043	82841000	27689.5
600	500	0.6	0.2	1.07548	1.16366	0.0105908	79399500	37337
700	500	0.6	0.15	1.06228	1.13003	0.00413123	78871600	46564.6
700	500	0.65	0.15	0.900575	0.820068	0.0180429	80803000	44616.3
750	600	0.65	0.15	1.15662	1.33835	0.0245639	78135000	13802.1
750	600	0.65	0.1	0.9685	0.939183	0.00113425	82193600	1590.02
800	600	0.75	0.1	1.11036	1.23297	0.0121805	80309600	13427.4
800	600	0.75	0.07	0.866586	0.750698	0.0178068	72469300	17882
1000	1000	0.75	0.1	0.968	0.93695	0.00102456	74334800	2325.65
1000	1200	0.75	0.1	0.968387	0.93764	0.00100116	74864300	12.1836
1200	1000	0.75	0.1	1.00067	1.00133	4.45697E-007	79457300	127.619

Tabla 47: Resultados para la función de Rosenbrock utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1200), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación(0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1.02606	1.05288	0.000679684	80801400	29369.8
19	0.989582	0.979249	0.000108592	75918500	900.155
43	1.06941	1.14243	0.00496206	83243500	204.582
83	1.01457	1.02935	0.000212186	79502400	686.298
101	1.03593	1.07311	0.00129099	82132800	251.761
137	0.98308	0.96642	0.000286352	81980200	4274.26
151	0.953179	0.908192	0.00220503	79552300	23.7067
191	1.02625	1.05296	0.000694804	80450900	32.4434
211	1.01768	1.03569	0.000312711	79057400	45.4474
233	0.988668	0.977396	0.000128888	84005400	3862.64
277	1.02216	1.04488	0.000491558	76866100	922.056
313	0.961455	0.924587	0.00148935	79158000	87.8421
349	0.987246	0.974818	0.000165289	79968000	5.23555
467	1.06299	1.12925	0.00401469	77176400	3085.98
521	0.984888	0.969876	0.000230029	80367500	21.0745
599	1.00688	1.01385	4.75337E-005	77160000	12577.6
617	0.995031	0.99015	2.50916E-005	80049300	9.06391
661	1.00586	1.01174	3.44393E-005	78660600	15159
683	1.00921	1.01852	8.48264E-005	82703800	19180.6
701	1.00804	1.0136	0.000717086	81270600	1896.74
727	1.02648	1.05381	0.000703523	81274000	1.74434
743	0.98951	0.979069	0.000110409	79405700	4380.83
797	1.00197	1.00395	3.8952E-006	79826900	837.199
811	0.957022	0.915939	0.00184729	78713200	117.944
853	1.02581	1.0532	0.000748142	84129200	2435.81
883	1.02619	1.05326	0.000689682	78109200	12369.8
907	0.993824	0.987622	3.85565E-005	81482800	451.89
991	0.990309	0.980756	9.41174E-005	81775700	7669.04

Tabla 48: Resultados para la función de Rosenbrock utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	3.8952E-006
Media de los resultados	0.000801
Desviación estándar de los resultados	0.001195

**B.1.19. Rosenbrock & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	4.74146	22.2806	18.0299	582868	5805.55
50	50	0.8	0.1	2.19262	4.80948	1.4227	59549500	14.2645
100	50	0.8	0.1	1.53414	2.35484	0.285461	49143700	9632.25
150	50	0.8	0.1	0.533757	0.425016	2.18072	77127100	490.794
150	150	0.8	0.1	0.65821	0.428255	0.119306	67483800	84.087
150	150	0.7	0.15	1.00723	1.03415	0.0386009	72419200	109.652
150	300	0.7	0.15	1.04357	1.08912	0.0018991	61429800	71083.4
250	300	0.7	0.15	0.739157	0.546702	0.068051	79088300	1567.14
350	300	0.7	0.15	0.993852	0.987924	4.11348E-005	72756600	56641.7
400	300	0.7	0.15	0.892393	0.803358	0.0164692	79713500	9.91894
450	300	0.7	0.15	1.2146	1.47551	0.0460599	74713800	46048.1
500	300	0.7	0.15	0.838033	0.694648	0.0320867	76636600	462.635
500	400	0.7	0.15	0.959704	0.916591	0.00359559	80061900	68.3537
500	400	0.6	0.15	1.00557	1.01062	6.18061E-005	77113100	78958.1
500	400	0.5	0.15	1.07552	1.15699	0.00570952	75548500	245.146
500	400	0.5	0.18	0.769756	0.588722	0.054458	82940800	1575.1
500	500	0.5	0.18	1.16587	1.36357	0.0293814	80603000	60062.6
600	500	0.6	0.18	1.11244	1.23804	0.0126693	81226200	3159.69
600	500	0.6	0.2	0.931984	0.868904	0.00463575	76223800	3243.43
700	500	0.6	0.15	0.974089	0.951877	0.00158764	77657300	43597.7
700	500	0.65	0.15	1.01927	1.03969	0.000430827	84321000	22956.4
750	600	0.65	0.15	1.00551	1.009	0.000452305	79419000	457.023
750	600	0.65	0.1	0.880331	0.774276	0.0143705	77109300	11956.3
800	600	0.75	0.1	1.08867	1.1859	0.00791081	78050300	15.4147
800	600	0.75	0.07	1.09732	1.20451	0.00948669	76583500	6802.83
1000	1000	0.75	0.1	1.11496	1.24475	0.013475	76796200	12945.4
1000	1200	0.75	0.1	1.11526	1.24294	0.0133582	76962700	8297.95
1200	1000	0.75	0.1	0.998303	0.996578	2.97192E-006	82495500	6469.75

Tabla 49: Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1200), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1.04756	1.09747	0.00226293	83705000	8.58346
19	0.98589	0.971479	0.000224166	81268400	118.202
43	0.991562	0.98318	7.12203E-005	82116400	35613.5
83	1.0513	1.1055	0.00263884	79605500	43375
101	0.983644	0.967477	0.000268137	79441200	3398.69
137	1.01636	1.03309	0.000268763	79969700	85.4544
151	0.980419	0.961067	0.000385801	78310400	1346.99
191	0.952667	0.907747	0.00224345	79295900	24158.1
211	0.938125	0.878902	0.00396701	77721000	1197.6
233	1.00911	1.01832	8.30053E-005	80289500	7718.21
277	0.96719	0.935483	0.00107658	81016600	19847.9
313	0.966769	0.934725	0.00110501	81246900	150.331
349	1.03228	1.06506	0.00107166	80930900	17.5153
467	1.03124	1.0637	0.000981769	80461400	1.65358
521	0.980322	0.960956	0.000387785	74395700	56.9834
599	1.04159	1.0849	0.0017295	81130000	600.111
617	1.08819	1.18352	0.0078192	79417400	38.0334
661	1.04236	1.08672	0.00179863	78190100	20972.7
683	1.00274	1.0055	7.53242E-006	81237900	1341.35
701	1.00428	1.0086	1.83624E-005	79342200	226.734
727	0.99725	0.994502	0.000007565	81341800	3196.89
743	0.999159	0.998316	0.000000708	78479500	33312.5
797	1.02475	1.05099	0.000688296	78727300	3723.13
811	0.971202	0.943285	0.00082956	83932700	1353.32
853	1.00816	1.01644	6.68933E-005	77941600	7665.98
883	0.972395	0.945512	0.00076219	76132400	16233.3
907	0.978568	0.957358	0.000464963	79252400	83.0916
991	1.02703	1.05441	0.000745166	82397800	3497.17

Tabla 50: Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	7.08E-007
Media de los resultados	0.001142
Desviación estándar de los resultados	0.001624

**B.1.20. Rosenbrock & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	1.56864	2.41836	0.501958	3260010	2532.47
50	50	0.8	0.1	0.519241	0.201046	0.701247	76319900	106.584
100	50	0.8	0.1	1.56894	2.43085	0.418158	53049500	6488.08
150	50	0.8	0.1	1.09612	1.16109	0.172355	70848700	1117.29
150	150	0.8	0.1	1.0889	1.18277	0.00876021	71824000	25327.6
150	150	0.7	0.15	1.44072	2.06562	0.204362	74667200	13924
150	300	0.7	0.15	1.15058	1.34199	0.0556799	74728200	8.43379
250	300	0.7	0.15	1.49723	2.24126	0.247258	74092100	11735.8
350	300	0.7	0.15	1.10396	1.21652	0.0112899	75204800	14224.1
400	300	0.7	0.15	0.880334	0.78011	0.0169444	78636300	521.709
450	300	0.7	0.15	0.652255	0.41796	0.126516	78244400	144.136
500	300	0.7	0.15	0.86852	0.759676	0.0201478	81873700	20214.4
500	400	0.7	0.15	0.867326	0.749033	0.0186398	82362400	53.9634
500	400	0.6	0.15	0.857141	0.733798	0.0204883	79212600	173.548
500	400	0.5	0.15	0.969976	0.941434	0.000935258	77059200	2156.4
500	400	0.5	0.18	0.942807	0.885216	0.00461681	71845600	265.914
500	500	0.5	0.18	0.942847	0.885471	0.00448412	77752400	16.3859
600	500	0.6	0.18	0.929986	0.865357	0.00492536	79891400	5539.3
600	500	0.6	0.2	1.01606	1.03254	0.00026066	77159800	68200.2
700	500	0.6	0.15	1.06476	1.13392	0.00419816	81618800	1288.18
700	500	0.65	0.15	0.997164	0.993925	2.49876E-005	77149400	50.4362
750	600	0.65	0.15	0.903875	0.807711	0.0178512	77667500	65907.1
750	600	0.65	0.1	0.98425	0.968551	0.000251983	82061600	796.495
800	600	0.75	0.1	1.03135	1.06368	0.000983116	77146400	1.11488
800	600	0.75	0.07	1.0112	1.02182	0.000176136	74560400	2590.79
1000	1000	0.75	0.1	0.953369	0.908844	0.0021749	76046800	586.393
1000	1200	0.75	0.1	1.02723	1.05324	0.00112661	80903100	15360.1
1200	1000	0.75	0.1	0.988204	0.976698	0.000141425	81100500	393.85

Tabla 51: Resultados para la función de Rosenbrock utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

7	0.919109	0.844943	0.00654654	79756200	2571.1
19	1.03401	1.06931	0.00115853	79048700	44118.2
43	0.960224	0.921632	0.00159802	78669100	69119.7
83	1.04243	1.08677	0.00180156	72786400	2161.53
101	0.974341	0.949309	0.000658496	75915600	3404.44
137	1.02018	1.04113	0.000420336	77610300	599.802
151	0.98918	0.978724	0.000123137	76659000	4.54844
191	0.961875	0.925213	0.00145356	79787500	140.199
211	0.98131	0.962391	0.000382696	79833000	0.586141
233	0.959117	0.919922	0.00167142	75661100	34496.1
277	0.998286	0.99763	0.000114101	76150500	1523.64
313	1.00186	1.00366	3.90474E-006	74817700	38328.6
349	0.803359	0.643186	0.0391514	83152200	183.473
467	1.0967	1.20261	0.00935402	74332800	921.443
521	0.981998	0.964247	0.00032461	78213200	516.672
599	1.06435	1.13281	0.00414064	78080500	843.464
617	1.14983	1.32216	0.0224504	78848500	0.0336197
661	1.01088	1.02184	0.000118582	77344700	25466.1
683	1.01119	1.02245	0.000125433	78453700	479.786
701	0.982704	0.965581	0.000300751	73383500	9210.85
727	0.979584	0.959491	0.000417692	77769000	12276.5
743	1.01103	1.02205	0.000123326	78890200	63208.8
797	0.995827	0.99639	0.00224428	80032200	28.4691
811	1.0704	1.14415	0.005215	79952600	136.746
853	1.07595	1.15841	0.00582354	77460500	21183.2
883	0.880654	0.774069	0.0144632	79812400	624.145
907	0.86635	0.750329	0.0178677	78236600	190.343
991	1.02205	1.04477	0.000489666	78049900	29276.8

Tabla 52: Resultados para la función de Rosenbrock utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	3.90474E-006
Media de los resultados	0.004948
Desviación estándar de los resultados	0.008833

**B.1.21. Schwefel & Cruza de 2 puntos**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	430.138	421.84	10.6978	855.619	42.7809
50	50	0.8	0.1	420.874	420.674	0.0120975	1343.24	8.39474
100	50	0.8	0.1	419.696	421.904	0.314962	1329.95	9.16774
150	50	0.8	0.1	420.862	421.194	0.00785351	1506.6	5.02945
150	150	0.8	0.1	420.996	420.994	0.000195583	1502.41	2.8053
150	150	0.7	0.15	421.193	420.951	0.00640292	1446.99	3.08461
150	300	0.7	0.15	420.962	420.964	3.36932E-005	1554.34	2.2912
250	300	0.7	0.15	420.967	421.186	0.0060083	1453.08	2.91811
350	300	0.7	0.15	420.948	420.985	0.000112779	1553.3	2.35587
400	300	0.7	0.15	420.999	420.978	0.000152205	1552.65	1.54069
450	300	0.7	0.15	421.144	421.433	0.0311197	1524.27	3.10344
500	300	0.7	0.15	421.01	420.832	0.00259378	1560.63	0.819369
500	400	0.7	0.15	420.97	420.832	0.00237763	1523.4	2.53136
500	400	0.6	0.15	420.971	421.02	0.000355706	1598.92	1.18626
500	400	0.5	0.15	420.962	420.975	3.55634E-005	1640.58	1.72375
500	400	0.5	0.18	420.942	420.988	0.000159181	1551.23	2.21195
500	500	0.5	0.18	420.942	420.984	0.000149817	1438.28	0.297806
600	500	0.6	0.18	421.056	420.929	0.00119074	1578.75	1.9566
600	500	0.6	0.2	420.953	421.001	0.000192283	1669.89	1.34698
700	500	0.6	0.15	420.956	420.969	4.73903E-005	1576.54	0.848594
700	500	0.65	0.15	420.989	420.957	9.50521E-005	1522	1.14847
750	600	0.65	0.15	420.987	420.939	0.000179305	1516.61	0.58391
750	600	0.65	0.1	420.968	420.969	2.55226E-005	1549.02	1.23975
800	600	0.75	0.1	420.967	420.97	2.59819E-005	1514.92	1.01115
800	600	0.75	0.07	420.967	420.97	2.59404E-005	1572.01	1.09218
1000	1000	0.75	0.1	420.969	420.969	2.54647E-005	1548.75	0.279193
1000	1200	0.75	0.1	420.969	420.969	2.54574E-005	1489.71	0.372252
1200	1000	0.75	0.1	420.97	420.969	2.55689E-005	1575.12	0.61748

Tabla 53: Resultados para la función de Schwefel utilizando Cruza de 2 puntos y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.969	420.969	2.54558E-005	1487.25	0.923355
19	420.968	420.968	2.55729E-005	1526.61	0.936899
43	420.971	420.966	2.69677E-005	1657.92	0.229611
83	420.968	420.969	2.54817E-005	1552.85	0.940264
101	420.969	420.969	2.54565E-005	1511.71	0.423695
137	420.968	420.969	2.54702E-005	1621.59	0.7957
151	420.969	420.969	2.54552E-005	1545.34	0.809638
191	420.969	420.969	2.54879E-005	1618.03	0.104073
211	420.969	420.969	2.54554E-005	1535.75	0.709253
233	420.971	420.967	2.64184E-005	1576.31	0.39949
277	420.969	420.968	2.55192E-005	1628.2	0.338665
313	420.968	420.968	2.54741E-005	1580.86	0.399818
349	420.97	420.967	2.61871E-005	1608.91	0.0709988
467	420.967	420.969	0.000025933	1580.51	0.865438
521	420.969	420.968	2.54938E-005	1541.31	0.429883
599	420.968	420.969	2.55395E-005	1530.65	0.911547
617	420.967	420.968	2.59717E-005	1625.53	0.637233
661	420.959	420.968	3.69821E-005	1517.35	0.930027
683	420.968	420.969	2.55184E-005	1539.7	0.555037
701	420.969	420.969	2.54774E-005	1509.1	1.27519
727	420.969	420.969	2.54664E-005	1542.09	1.05389
743	420.969	420.969	2.54553E-005	1542.22	0.985265
797	420.968	420.969	2.54853E-005	1603.19	0.74808
811	420.969	420.969	2.55109E-005	1598.11	0.847713
853	420.968	420.969	2.54785E-005	1560.54	1.1138
883	420.968	420.968	2.54929E-005	1517.18	0.349718
907	420.969	420.968	2.55241E-005	1660.46	0.957577
991	420.969	420.969	2.54569E-005	1536.42	0.401778

Tabla 54: Resultados para la función de Schwefel utilizando Cruza de 2 puntos y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54552E-005
Media de los resultados	2.604244E-005
Desviación estándar de los resultados	2.173102E-006

**B.1.22. Schwefel & PAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	417.195	418.994	2.28469	729.175	29.6186
50	50	0.8	0.1	419.216	417.801	1.65089	1250.55	8.74595
100	50	0.8	0.1	421.62	421.124	0.0565902	1195.25	11.6213
150	50	0.8	0.1	421.992	419.943	0.265001	1442.11	5.76902
150	150	0.8	0.1	420.63	420.787	0.018616	1513.36	7.0304
150	150	0.7	0.15	420.928	420.929	0.000434477	1411.44	5.55413
150	300	0.7	0.15	420.977	420.955	0.000056187	1367.04	6.63011
250	300	0.7	0.15	420.92	420.96	0.000336858	1515.58	4.93186
350	300	0.7	0.15	420.977	420.968	3.42807E-005	1551.9	2.00737
400	300	0.7	0.15	420.855	420.832	0.00400859	1615.34	1.03447
450	300	0.7	0.15	420.967	420.98	4.28354E-005	1437.6	2.51675
500	300	0.7	0.15	421.139	420.892	0.00444545	1537.68	2.41413
500	400	0.7	0.15	420.934	420.954	0.000206396	1526.57	1.01958
500	400	0.6	0.15	421.006	420.967	0.000197118	1525.97	1.72395
500	400	0.5	0.15	421.029	421.037	0.00107835	1552.22	0.446651
500	400	0.5	0.18	420.946	421.023	0.000464765	1537.25	1.67299
500	500	0.5	0.18	420.946	421.023	0.000464765	1532.08	2.03723
600	500	0.6	0.18	420.948	420.96	8.94469E-005	1669.1	1.30115
600	500	0.6	0.2	421.024	420.893	0.00113959	1605.34	1.47995
700	500	0.6	0.15	420.979	421.025	0.0004368	1566.62	1.0484
700	500	0.65	0.15	421.009	420.971	0.000234426	1522.25	0.781115
750	600	0.65	0.15	420.965	420.938	0.000149643	1500.44	0.645866
750	600	0.65	0.1	420.968	420.97	2.56661E-005	1532.85	1.14957
800	600	0.75	0.1	420.968	420.975	3.07018E-005	1642.25	0.804261
800	600	0.75	0.07	420.969	420.969	0.000025469	1566.77	0.370156
1000	1000	0.75	0.1	420.968	420.969	2.55355E-005	1644.71	0.620263
1000	1200	0.75	0.1	420.968	420.969	2.55295E-005	1510.81	0.725602
1200	1000	0.75	0.1	420.969	420.959	3.64533E-005	1552.52	0.452791

Tabla 55: Resultados para la función de Schwefel utilizando PAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.969	420.969	2.54552E-005	1535.01	0.49848
19	420.969	420.969	2.54559E-005	1476.94	1.07103
43	420.969	420.969	2.54699E-005	1491.51	1.00848
83	420.969	420.969	2.54572E-005	1508.09	0.729912
101	420.969	420.969	2.54588E-005	1424.85	1.20619
137	420.969	420.969	2.54555E-005	1590.6	0.728128
151	420.969	420.969	2.54585E-005	1643.25	0.577931
191	420.969	420.969	2.54577E-005	1557.14	0.828794
211	420.969	420.968	2.55554E-005	1574.18	0.39645
233	420.969	420.969	2.54553E-005	1568.2	0.178704
277	420.969	420.969	2.54551E-005	1648.46	0.352621
313	420.969	420.969	2.54552E-005	1578.13	0.32704
349	420.968	420.969	2.54732E-005	1501.78	0.507254
467	420.969	420.968	0.000025485	1499.32	1.47056
521	420.969	420.969	2.54684E-005	1502.29	0.0223268
599	420.969	420.969	2.54552E-005	1563.24	1.1621
617	420.969	420.969	2.54555E-005	1556.72	0.707338
661	420.97	420.968	2.58097E-005	1546.14	0.185352
683	420.969	420.969	2.54566E-005	1412.84	1.16183
701	420.969	420.968	2.55166E-005	1642.34	0.647582
727	420.969	420.969	2.54644E-005	1646.79	1.0102
743	420.969	420.969	2.54552E-005	1547.38	0.321785
797	420.969	420.969	2.54554E-005	1654.34	0.670875
811	420.969	420.969	2.54553E-005	1555.25	0.950405
853	420.969	420.969	2.54555E-005	1500.92	0.543223
883	420.969	420.969	2.54568E-005	1519.74	0.769913
907	420.971	420.969	2.60765E-005	1607.12	0.186809
991	420.969	420.969	2.54552E-005	1497.18	0.522742

Tabla 56: Resultados para la función de Schwefel utilizando PAX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54551E-005
Media de los resultados	2.549943E-005
Desviación estándar de los resultados	1.322777E-007

**B.1.23. Schwefel & SANUX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	415.679	419.38	3.837	1135.61	36.4118
50	50	0.8	0.1	420.941	420.706	0.00884893	1370.26	15.8442
100	50	0.8	0.1	421	421.039	0.000781348	1342.25	4.15024
150	50	0.8	0.1	422.854	421.298	0.462388	1494.5	8.61408
150	150	0.8	0.1	420.95	420.968	7.08152E-005	1506.68	5.72915
150	150	0.7	0.15	421.166	421.329	0.0213727	1266.53	0.000142485
150	300	0.7	0.15	421.052	420.84	0.00297619	1504.02	4.64138
250	300	0.7	0.15	421.014	420.942	0.000370202	1422.64	1.52068
350	300	0.7	0.15	421.045	420.994	0.000845581	1520.91	2.1854
400	300	0.7	0.15	421.019	420.666	0.0119194	1536.09	1.88544
450	300	0.7	0.15	420.943	420.966	0.000106848	1662.31	1.96218
500	300	0.7	0.15	420.99	420.921	0.000371441	1481.54	0.465686
500	400	0.7	0.15	420.99	420.921	0.000371441	1551.17	2.18822
500	400	0.6	0.15	420.997	420.943	0.000212654	1541.6	1.91069
500	400	0.5	0.15	420.999	421.153	0.00441267	1646.6	1.45265
500	400	0.5	0.18	421.106	420.965	0.00240255	1557.08	1.27889
500	500	0.5	0.18	421.049	420.985	0.000877139	1537.36	0.773534
600	500	0.6	0.18	421.052	420.996	0.00100194	1580.82	0.908118
600	500	0.6	0.2	421.358	420.949	0.0191814	1560.12	1.24176
700	500	0.6	0.15	420.965	420.973	2.93941E-005	1497.67	1.03752
700	500	0.65	0.15	420.964	421.055	0.000956435	1547.17	1.78568
750	600	0.65	0.15	420.969	420.975	3.03556E-005	1559.37	1.1763
750	600	0.65	0.1	420.967	420.972	2.72408E-005	1534.88	1.01839
800	600	0.75	0.1	420.969	420.968	2.55254E-005	1567.66	1.00749
800	600	0.75	0.07	420.969	420.969	2.54551E-005	1536.89	0.849623
1000	1000	0.75	0.1	420.967	420.969	2.56927E-005	1606.07	1.0363
1000	1200	0.75	0.1	420.967	420.969	2.56668E-005	1484.27	0.860078
1200	1000	0.75	0.1	420.969	420.97	0.000025528	1546.43	0.58354

Tabla 57: Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.969	420.969	2.54555E-005	1655.82	0.875745
19	420.968	420.967	2.57008E-005	1565.98	0.744116
43	420.969	420.969	2.54645E-005	1644.73	0.658144
83	420.969	420.972	2.71522E-005	1552.86	1.04237
101	420.968	420.969	2.55047E-005	1586.12	0.783724
137	420.969	420.969	2.54552E-005	1523.32	0.165827
151	420.97	420.962	0.00003084	1571.34	1.11112
191	420.969	420.969	2.54569E-005	1609.69	0.47338
211	420.968	420.97	2.58428E-005	1521.63	1.03127
233	420.969	420.969	2.54555E-005	1596.2	1.59155
277	420.968	420.969	0.00002561	1603.29	1.03934
313	420.968	420.969	2.54712E-005	1544.95	1.55089
349	420.969	420.969	2.54585E-005	1547.03	0.757625
467	420.969	420.969	2.54557E-005	1555.33	1.4773
521	420.967	420.968	2.59006E-005	1544.51	0.829402
599	420.968	420.969	2.56235E-005	1518.96	1.15623
617	420.969	420.969	0.000025458	1538.05	0.470667
661	420.968	420.966	2.64921E-005	1641.75	0.817398
683	420.969	420.968	2.55266E-005	1491.35	3.19082E-008
701	420.969	420.968	2.54831E-005	1536.6	1.07993
727	420.968	420.971	2.60235E-005	1640.53	1.54136
743	420.969	420.969	2.54553E-005	1612.03	1.64274
797	420.968	420.969	2.55147E-005	1495.95	1.2556
811	420.969	420.969	0.000025457	1633.64	1.01117
853	420.97	420.969	2.58158E-005	1519.28	0.702114
883	420.968	420.975	3.01252E-005	1669.98	0.48893
907	420.968	420.969	2.55539E-005	1435.19	0.296152
991	420.969	420.969	2.54602E-005	1529.5	0.284768

Tabla 58: Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54552E-005
Media de los resultados	2.600761E-005
Desviación estándar de los resultados	1.321501E-006

**B.1.24. Schwefel & Selective Crossover**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	430.156	422.546	10.9568	910.933	20.3696
50	50	0.8	0.1	418.839	419.894	0.717532	1309.58	18.3698
100	50	0.8	0.1	420.078	420.745	0.106458	1364.89	9.2311
150	50	0.8	0.1	420.732	421.057	0.00809788	1338.14	0.14292
150	150	0.8	0.1	421.011	420.946	0.000316984	1467.05	6.3423
150	150	0.7	0.15	420.402	421.949	0.161818	1406.22	4.0214
150	300	0.7	0.15	420.792	421.4	0.0274474	1434.33	2.85123
250	300	0.7	0.15	420.963	421.032	0.000528848	1420.45	3.74757
350	300	0.7	0.15	420.811	421.157	0.00762686	1455.88	0.899015
400	300	0.7	0.15	420.832	421.146	0.00634636	1554.61	1.12361
450	300	0.7	0.15	421.032	420.982	0.000557478	1506.18	2.11429
500	300	0.7	0.15	421.047	420.932	0.000979889	1446.36	1.31304
500	400	0.7	0.15	420.992	420.983	0.000120346	1390.37	2.3137
500	400	0.6	0.15	421	420.951	0.000186072	1516.68	2.08344
500	400	0.5	0.15	420.917	421.003	0.000504599	1462.09	1.94512
500	400	0.5	0.18	421.035	420.837	0.00276619	1587.34	2.03321
500	500	0.5	0.18	421.006	420.974	0.000204851	1495.75	1.39533
600	500	0.6	0.18	421.334	421.017	0.0171445	1545.47	1.40361
600	500	0.6	0.2	420.909	420.946	0.000535943	1471.87	1.44961
700	500	0.6	0.15	420.967	420.974	2.94667E-005	1585.44	0.962184
700	500	0.65	0.15	420.932	420.955	0.000216874	1653.95	1.21295
750	600	0.65	0.15	420.939	421.011	0.000367486	1520.48	0.589275
750	600	0.65	0.1	420.969	420.968	2.54662E-005	1601.32	0.68742
800	600	0.75	0.1	420.97	420.967	0.000026096	1551.14	1.22614
800	600	0.75	0.07	420.969	420.969	2.54608E-005	1490.56	1.0617
1000	1000	0.75	0.1	420.969	420.969	0.000025463	1647.12	0.0115512
1000	1200	0.75	0.1	420.969	420.969	2.54596E-005	1538.71	0.790409
1200	1000	0.75	0.1	420.969	420.969	2.54707E-005	1658.49	0.626275

Tabla 59: Resultados para la función de Schwefel utilizando Selective Crossover y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.968	420.968	0.000025489	1575.58	0.454641
19	420.967	420.969	2.59411E-005	1554.31	0.899744
43	420.967	420.968	2.61295E-005	1542.78	0.352331
83	420.968	420.968	2.55924E-005	1523.93	0.979247
101	420.968	420.969	2.55033E-005	1524.94	1.30831
137	420.971	420.968	2.62867E-005	1629.99	0.751647
151	420.969	420.97	2.55303E-005	1552.99	0.694462
191	420.971	420.969	2.60303E-005	1526.66	0.120462
211	420.968	420.968	0.000025602	1591.38	0.919314
233	420.97	420.97	0.0000259	1464.96	0.441316
277	420.967	420.97	2.58463E-005	1519.06	0.603413
313	420.969	420.969	2.54797E-005	1588.93	0.179362
349	420.969	420.969	2.54613E-005	1490.59	0.242775
467	420.968	420.969	2.54642E-005	1615.74	0.588273
521	420.97	420.969	2.55414E-005	1644.96	0.90854
599	420.969	420.969	2.54559E-005	1646.45	0.755266
617	420.969	420.968	2.54971E-005	1534.8	0.347617
661	420.969	420.969	2.54663E-005	1649.07	0.550451
683	420.969	420.969	2.54564E-005	1634.87	0.425611
701	420.969	420.969	2.54563E-005	1574.48	1.27046
727	420.969	420.966	0.000026691	1612.71	0.846776
743	420.969	420.969	0.000025459	1599.89	1.22215
797	420.969	420.969	2.54718E-005	1497.03	0.679406
811	420.968	420.968	2.55125E-005	1519.69	0.534328
853	420.97	420.969	2.56524E-005	1624.92	0.52719
883	420.969	420.969	2.54569E-005	1498.83	0.647399
907	420.967	420.969	2.58651E-005	1539.84	0.57889
991	420.968	420.967	2.58579E-005	1552.08	0.818848

Tabla 60: Resultados para la función de Schwefel utilizando SANUX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54559E-005
Media de los resultados	0.000025
Desviación estándar de los resultados	3.09285E-007

## B.2. Representación real

### B.2.1. Ackley & GPAX

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.157924	0.721624	3.49933	21.5143	4.22511
50	50	0.8	0.1	-0.0368995	-0.00149951	0.140364	20.693	2.02702
100	50	0.8	0.1	-0.160742	0.0119592	1.02105	22.277	4.86098
150	50	0.8	0.1	0.0027446	-0.0265104	0.0941891	22.1346	4.1781
150	150	0.8	0.1	1.63374E-005	-0.0265299	0.0936713	21.9502	3.22106
150	150	0.7	0.15	-8.28365E-006	5.01848E-005	0.000143934	21.986	4.95482
150	300	0.7	0.15	-1.03874E-005	1.00261E-005	4.08388E-005	22.2302	5.12578
250	300	0.7	0.15	2.54811E-006	-0.00497533	0.0147314	22.208	5.71501
350	300	0.7	0.15	-0.00192192	-0.00318092	0.0108795	22.3091	6.95818
400	300	0.7	0.15	-0.0027156	-0.00148626	0.00901119	22.1804	6.55001
450	300	0.7	0.15	-4.26997E-005	-4.91029E-005	0.000184164	22.2993	6.12485
500	300	0.7	0.15	0.000316681	-0.00209529	0.00611324	22.288	6.32666
500	400	0.7	0.15	2.97159E-005	-0.00210401	0.00606954	22.282	6.36102
500	400	0.6	0.15	-0.00150386	-0.000267139	0.00438226	22.2779	6.7422
500	400	0.5	0.15	-0.0290358	-0.0511605	0.256323	22.3067	8.84446
500	400	0.5	0.18	-0.000005313	-9.73891E-006	3.13815E-005	22.3322	8.57685
500	500	0.5	0.18	-0.000005313	-9.73891E-006	3.13815E-005	22.3439	9.91691
600	500	0.6	0.18	-0.0215497	-0.00588471	0.07642	22.3267	8.9749
600	500	0.6	0.2	-5.76363E-006	0.000000208	1.63135E-005	22.2704	6.71635
700	500	0.6	0.15	0.000217937	-0.00676595	0.0203667	22.3321	7.60967
700	500	0.65	0.15	-0.00153207	6.18309E-006	0.00439588	22.3057	7.48589
750	600	0.65	0.15	0	0	-4.44089E-016	22.338	7.51174
750	600	0.65	0.1	0	0	-4.44089E-016	22.321	5.57891
800	600	0.75	0.1	-0.00010232	-0.00428691	0.0126182	22.3284	5.8127
800	600	0.75	0.07	-0.000412852	-0.00029323	0.00143912	22.339	5.37716
1000	1000	0.75	0.1	-0.0045001	-0.00289475	0.0158965	22.3447	5.98944
1000	1200	0.75	0.1	-0.0045001	-0.00289475	0.0158965	22.3334	5.34111
1200	1000	0.75	0.1	-3.52621E-006	-0.00537122	0.0159601	22.3456	5.4454

Tabla 61: Resultados para la función de Ackley utilizando GPAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0	0	-4.44089E-016	22.3342	6.54326
19	0	0	-4.44089E-016	22.3309	6.97931
43	-5.95986E-005	-0.0121518	0.0382985	22.3127	4.90705
83	0	0	-4.44089E-016	22.3456	7.06325
101	-0.0057427	-0.0145612	0.0507849	22.1903	4.45479
137	0	0	-4.44089E-016	22.2769	7.04574
151	0	0	-4.44089E-016	22.3442	6.86642
191	0	0	-4.44089E-016	22.343	5.85581
211	0	0	-4.44089E-016	22.3077	6.17652
233	0	0	-4.44089E-016	22.3055	6.53051
277	0	0	-4.44089E-016	22.3414	6.54558
313	0	0	-4.44089E-016	22.3165	6.32064
349	0	0	-4.44089E-016	22.3059	6.04449
467	0	0	-4.44089E-016	22.3351	6.95742
521	0	0	-4.44089E-016	22.2798	5.01036
599	0	0	-4.44089E-016	22.3107	7.18253
617	0	0	-4.44089E-016	22.333	6.90754
661	0	0	-4.44089E-016	22.2355	4.91297
683	0	0	-4.44089E-016	22.281	6.59869
701	0	0	-4.44089E-016	22.3305	7.439
727	0	0	-4.44089E-016	22.3459	6.37007
743	0	0	-4.44089E-016	22.2747	3.90351
797	0	0	-4.44089E-016	22.3366	4.86189
811	0	0	-4.44089E-016	22.2938	7.12735
853	0	0	-4.44089E-016	22.3211	7.91009
883	0	0	-4.44089E-016	22.3242	7.01184
907	0	0	-4.44089E-016	22.3304	6.08986
991	0	0	-4.44089E-016	22.3064	6.73677

Tabla 62: Resultados para la función de Ackley utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	-4.44089E-016
Media de los resultados	0.003071
Desviación estándar de los resultados	0.011804

**B.2.2. Ackley & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-1.89186	1.93917	6.73742	20.409	7.5484
50	50	0.8	0.1	0.989981	0.981241	3.59032	20.0413	4.86845
100	50	0.8	0.1	-0.0013273	0.000373238	0.0039504	20.2445	1.92986
150	50	0.8	0.1	-0.000454657	-0.0183015	0.0606803	21.4223	2.50875
150	150	0.8	0.1	-0.000454657	-0.0183015	0.0606803	20.0542	1.73189
150	150	0.7	0.15	0.99612	0.0242691	2.64485	20.0819	4.80331
150	300	0.7	0.15	0.99612	0.0242691	2.64485	20.0213	4.95683
250	300	0.7	0.15	0.953147	-0.00197133	2.58005	20.0263	4.95376
350	300	0.7	0.15	-0.000357712	0.00152671	0.00450062	20.0789	2.77421
400	300	0.7	0.15	0.00346628	-0.00323746	0.0140142	20.0339	2.79404
450	300	0.7	0.15	0.00496687	-0.000624346	0.0148262	20.0675	2.91404
500	300	0.7	0.15	0.0055324	0.00535972	0.0233663	20.1377	3.05576
500	400	0.7	0.15	0.0055324	0.00535972	0.0233663	20.073	2.92984
500	400	0.6	0.15	0.00473014	-0.000801068	0.0141821	20.1057	3.06016
500	400	0.5	0.15	0.00783803	0.0120445	0.0461362	20.2332	3.2146
500	400	0.5	0.18	0.00633733	0.0173899	0.061449	22.1496	4.14945
500	500	0.5	0.18	0.00633733	0.0173899	0.061449	22.017	3.53594
600	500	0.6	0.18	-0.000541743	0.00215707	0.0064223	20.0251	3.12106
600	500	0.6	0.2	-1.24447E-005	-0.000561059	0.00159569	20.0376	3.80811
700	500	0.6	0.15	-3.20881E-005	0.000878608	0.00250732	20.0513	2.85304
700	500	0.65	0.15	0	0	-4.44089E-016	20.0526	2.92977
750	600	0.65	0.15	0	0	-4.44089E-016	20.0208	2.77085
750	600	0.65	0.1	0	0	-4.44089E-016	20.003	1.86667
800	600	0.75	0.1	0.00042568	-0.0012164	0.0036893	20.0158	1.91788
800	600	0.75	0.07	0.000134279	0.000134761	0.000539043	20.006	1.33903
1000	1000	0.75	0.1	0.000292725	-0.000316229	0.00122376	20.0145	1.92467
1000	1200	0.75	0.1	0.000292725	-0.000316229	0.00122376	20.0089	1.90826
1200	1000	0.75	0.1	0.000101878	-0.00115413	0.00331282	20.0118	1.95883

Tabla 63: Resultados para la función de Ackley utilizando Recombinación intermedia y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.000551151	0.000511646	0.00214212	20.0087	1.91621
19	0	0	-4.44089E-016	20.0241	2.0353
43	0	0	-4.44089E-016	20.0127	1.89869
83	0	0	-4.44089E-016	20.0007	1.93346
101	-2.26497E-005	0.00153473	0.00440409	20.0448	2.10922
137	0	0	-4.44089E-016	20.0125	2.01091
151	-6.23998E-005	0.000498632	0.00142807	20.013	2.00575
191	0	0	-4.44089E-016	20.0104	1.99975
211	0	0	-4.44089E-016	20.0061	1.90452
233	0	0	-4.44089E-016	20.0003	1.84823
277	0	0	-4.44089E-016	20.0014	1.96465
313	0	0	-4.44089E-016	20.0095	2.02478
349	0.000584102	-0.000518443	0.00222524	20.0142	2.03343
467	0	0	-4.44089E-016	20.0106	1.9212
521	0	0	-4.44089E-016	20.0214	1.9762
599	0	0	-4.44089E-016	20.0373	2.00791
617	0	0	-4.44089E-016	20.002	1.86169
661	0	0	-4.44089E-016	20.0085	1.8898
683	0	0	-4.44089E-016	20.0045	1.89959
701	0	0	-4.44089E-016	20.0032	1.92152
727	0	0	-4.44089E-016	20.0247	1.98401
743	0	0	-4.44089E-016	20.0187	1.9155
797	0	0	-4.44089E-016	20.0016	1.94061
811	3.19836E-005	-1.02001E-005	9.49824E-005	20.0036	1.94915
853	0	0	-4.44089E-016	20.0252	1.95334
883	0	0	-4.44089E-016	20.0194	1.99705
907	0	0	-4.44089E-016	20.0173	1.91374
991	0	0	-4.44089E-016	20.0122	1.95357

Tabla 64: Resultados para la función de Ackley utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	-4.44089E-016
Media de los resultados	0.0003676608
Desviación estándar de los resultados	0.0009945194

**B.2.3. Ackley & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.15	-0.987028	4.51015	11.3085	21.7189	11.9482
50	50	0.8	0.1	0.00135629	-0.9399	2.58347	21.409	4.79126
100	50	0.8	0.1	0.0191603	-0.0288931	0.129801	20.2681	2.53983
150	50	0.8	0.1	-0.00417583	0.0102043	0.0344196	21.7047	3.42127
150	150	0.8	0.1	6.36102E-005	3.20735E-005	0.000201629	20.0415	1.72779
150	150	0.7	0.15	-9.14043E-006	4.67537E-005	0.000134803	20.5807	2.78241
150	300	0.7	0.15	-1.39564E-005	4.38161E-005	0.000130122	20.4729	2.76503
250	300	0.7	0.15	6.90449E-005	-8.28883E-005	0.000305435	20.1739	2.84436
350	300	0.7	0.15	1.63837E-006	-3.8333E-007	4.75924E-006	20.7046	2.78106
400	300	0.7	0.15	3.53097E-005	-7.56189E-005	0.000236236	20.3483	2.86648
450	300	0.7	0.15	5.95321E-005	-0.000126929	0.000397058	22.3143	3.36998
500	300	0.7	0.15	8.10237E-005	-0.00015837	0.000504	22.2234	3.4366
500	400	0.7	0.15	5.38757E-005	-1.43485E-005	0.000157778	21.2964	3.13176
500	400	0.6	0.15	-1.29669E-005	2.49772E-005	7.96201E-005	21.4925	3.58741
500	400	0.5	0.15	-8.20166E-005	-0.000203359	0.000621483	21.6634	3.28244
500	400	0.5	0.18	-1.01391E-006	-2.54933E-006	7.76015E-006	22.0695	3.90697
500	500	0.5	0.18	-1.01391E-006	-2.54933E-006	7.76015E-006	22.1711	4.13942
600	500	0.6	0.18	8.89247E-005	-7.48594E-006	0.000252619	21.9933	3.67057
600	500	0.6	0.2	2.69984E-005	2.88828E-005	0.000111868	22.2078	4.74851
700	500	0.6	0.15	9.68636E-006	-0.00000518	3.10719E-005	21.9156	3.39077
700	500	0.65	0.15	0	0	-4.44089E-016	20.5262	2.92224
750	600	0.65	0.15	0	0	-4.44089E-016	21.5931	3.04959
750	600	0.65	0.1	0	0	-4.44089E-016	22.2827	2.28404
800	600	0.75	0.1	-8.62808E-006	-4.35144E-006	2.73343E-005	20.6523	2.09438
800	600	0.75	0.07	9.83764E-005	-0.000087515	0.000372879	20.2199	1.46522
1000	1000	0.75	0.1	1.44202E-005	-6.77043E-006	4.50651E-005	20.8336	2.14595
1000	1200	0.75	0.1	1.44202E-005	-6.77043E-006	4.50651E-005	21.1194	2.06315
1200	1000	0.75	0.1	1.38474E-005	-1.68495E-005	6.16995E-005	22.0239	2.28089

Tabla 65: Resultados para la función de Ackley utilizando SBX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-4.58491E-005	7.04897E-005	0.000238028	20.6541	2.35682
19	0	0	-4.44089E-016	22.2795	2.42983
43	0	0	-4.44089E-016	21.3097	2.1411
83	0	0	-4.44089E-016	21.2362	2.27725
101	5.34162E-005	-0.000033079	0.000177813	22.0313	2.48971
137	0	0	-4.44089E-016	21.1553	2.54723
151	2.57413E-005	-8.02341E-006	7.62815E-005	20.9765	2.32112
191	0	0	-4.44089E-016	22.0136	2.38385
211	0	0	-4.44089E-016	22.1814	2.47002
233	0	0	-4.44089E-016	21.6541	2.18399
277	0	0	-4.44089E-016	21.2236	2.29322
313	0	0	-4.44089E-016	20.4891	2.33519
349	1.77907E-005	-2.90903E-005	9.64781E-005	22.3289	2.58113
467	0	0	-4.44089E-016	20.6941	2.19323
521	0	0	-4.44089E-016	21.1512	2.29217
599	0	0	-4.44089E-016	22.185	2.46024
617	0	0	-4.44089E-016	22.3445	2.50054
661	0	0	-4.44089E-016	20.6538	2.05402
683	0	0	-4.44089E-016	21.439	2.19061
701	0	0	-4.44089E-016	22.0417	2.23581
727	0	0	-4.44089E-016	20.4289	2.19534
743	0	0	-4.44089E-016	21.9696	2.47425
797	0	0	-4.44089E-016	21.1831	2.22712
811	5.52777E-006	-4.05024E-005	0.000115665	21.525	2.39184
853	0	0	-4.44089E-016	21.0081	2.20734
883	0	0	-4.44089E-016	20.8034	2.38517
907	0	0	-4.44089E-016	20.9099	2.2249
991	0	0	-4.44089E-016	20.5597	2.36321

Tabla 66: Resultados para la función de Ackley utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido -4.44089E-016  
 Media de los resultados 2.515234E-005  
 Desviación estándar de los resultados 6.0573232E-005

**B.2.4. Ackley & SBX adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.0169163	0.0685158	0.327161	20.1536	1.66185
50	50	0.8	0.1	-0.0133626	0.0918208	0.476463	20.6941	2.45423
100	50	0.8	0.1	-0.0354709	-0.182705	1.23852	21.412	3.66208
150	50	0.8	0.1	0.00234296	-0.00953466	0.0303354	20.3194	1.9621
150	150	0.8	0.1	0.00309608	-0.00152566	0.0100797	20.0139	1.67207
150	150	0.7	0.15	-0.000826778	0.000139023	0.00239003	20.0483	2.43057
150	300	0.7	0.15	-0.000239724	0.000264596	0.00101326	20.0001	2.67157
250	300	0.7	0.15	3.57811E-008	-5.4629E-008	1.84708E-007	20.0039	2.72629
350	300	0.7	0.15	9.06665E-006	-3.58854E-007	2.56666E-005	20.0151	2.65214
400	300	0.7	0.15	-2.87531E-008	1.33415E-007	0.000000386	20.0415	2.71546
450	300	0.7	0.15	-8.45764E-006	-1.67376E-006	2.43877E-005	20.0294	2.80381
500	300	0.7	0.15	9.75706E-008	1.66699E-008	0.00000028	20.0591	2.77602
500	400	0.7	0.15	-8.42331E-009	5.12989E-009	2.78952E-008	20.0042	2.76294
500	400	0.6	0.15	5.13574E-007	-3.19168E-007	1.71028E-006	20.0348	2.87665
500	400	0.5	0.15	9.16433E-008	-9.8037E-007	0.000002785	20.0647	2.67084
500	400	0.5	0.18	-7.46337E-008	7.37947E-007	2.09789E-006	20.0483	3.22787
500	500	0.5	0.18	-1.1784E-007	4.85387E-007	1.41277E-006	20.0015	3.24831
600	500	0.6	0.18	-0.000000018	1.01085E-007	2.90395E-007	20.0155	3.04789
600	500	0.6	0.2	-2.36243E-007	1.01223E-007	0.000000727	20.0105	3.72275
700	500	0.6	0.15	-0.000000724	4.91069E-007	2.47443E-006	21.7552	3.13064
700	500	0.65	0.15	0	0	-4.44089E-016	20.0047	2.80552
750	600	0.65	0.15	0	0	-4.44089E-016	20.0101	2.70401
750	600	0.65	0.1	0	0	-4.44089E-016	20.0006	1.84415
800	600	0.75	0.1	4.37631E-007	-0.000002091	6.04243E-006	20.069	1.8941
800	600	0.75	0.07	0.000000155	1.62715E-007	6.35661E-007	20.051	1.32871
1000	1000	0.75	0.1	-1.84173E-008	6.44446E-009	5.51891E-008	20.0048	1.86302
1000	1200	0.75	0.1	-9.12562E-009	2.34843E-009	2.66521E-008	20.0001	1.86133
1200	1000	0.75	0.1	-2.16756E-008	-2.83317E-007	8.03686E-007	20.003	1.90993

Tabla 67: Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	9.95813E-008	-5.0559E-008	3.15882E-007	20.0583	1.89056
19	0	0	-4.44089E-016	20.0081	1.96394
43	0	0	-4.44089E-016	20.0165	1.85612
83	0	0	-4.44089E-016	20.0057	1.95857
101	-1.78938E-006	1.56736E-006	6.72831E-006	20.0215	2.01073
137	0	0	-4.44089E-016	20.0004	1.95022
151	5.18978E-008	-3.18271E-007	9.12099E-007	20.0152	1.97111
191	0	0	-4.44089E-016	20.0108	1.96945
211	0	0	-4.44089E-016	20.0086	1.89057
233	0	0	-4.44089E-016	20.0074	1.84788
277	0	0	-4.44089E-016	20.0109	1.93574
313	0	0	-4.44089E-016	20.0071	1.99551
349	2.6243E-007	-2.21122E-007	9.70629E-007	20.0693	2.00147
467	0	0	-4.44089E-016	20.0181	1.89526
521	0	0	-4.44089E-016	20.0193	1.91241
599	0	0	-4.44089E-016	20.103	1.939
617	0	0	-4.44089E-016	20.0017	1.84365
661	0	0	-4.44089E-016	20.001	1.84388
683	0	0	-4.44089E-016	20.0046	1.87225
701	0	0	-4.44089E-016	20.0009	1.89649
727	0	0	-4.44089E-016	20.0156	1.90946
743	0	0	-4.44089E-016	20.0246	1.86252
797	0	0	-4.44089E-016	20.0012	1.92431
811	-1.3895E-008	-1.56222E-008	5.91354E-008	20.0035	1.93141
853	0	0	-4.44089E-016	20.0018	1.87384
883	0	0	-4.44089E-016	20.001	1.92603
907	0	0	-4.44089E-016	20.0022	1.84945
991	0	0	-4.44089E-016	20.0037	1.90486

Tabla 68: Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido -4.44089E-016  
 Media de los resultados 3.209305E-007  
 Desviación estándar de los resultados 1.280387E-006

**B.2.5. Ackley & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.00159335	-0.00228011	0.00807377	20.0002	1.00987
50	50	0.8	0.1	3.81867E-011	4.22501E-012	1.08666E-010	19.9999	1.6
100	50	0.8	0.1	-4.35794E-013	-1.011E-011	2.86202E-011	19.9999	1.60021
150	50	0.8	0.1	-2.58138E-011	-1.14958E-011	7.9925E-011	19.9999	1.73334
150	150	0.8	0.1	1.49676E-017	-1.79533E-016	-4.44089E-016	19.9999	1.73333
150	150	0.7	0.15	3.44486E-017	-1.60541E-016	-4.44089E-016	19.9999	2.66666
150	300	0.7	0.15	3.44486E-017	-1.60541E-016	-4.44089E-016	19.9999	2.79999
250	300	0.7	0.15	1.1008E-016	3.28823E-016	-4.44089E-016	19.9999	2.87999
350	300	0.7	0.15	-8.01291E-017	2.35441E-016	-4.44089E-016	20	2.74285
400	300	0.7	0.15	-1.46886E-016	2.75105E-016	-4.44089E-016	19.9999	2.89999
450	300	0.7	0.15	-2.29179E-016	-5.78192E-017	-4.44089E-016	20	2.79999
500	300	0.7	0.15	1.01854E-017	-9.23559E-017	-4.44089E-016	20	2.67999
500	400	0.7	0.15	1.01854E-017	-9.23559E-017	-4.44089E-016	20	2.83999
500	400	0.6	0.15	1.79023E-016	-2.15853E-016	-4.44089E-016	20	2.67999
500	400	0.5	0.15	-1.50406E-017	2.24898E-016	-4.44089E-016	20	2.79999
500	400	0.5	0.18	3.52804E-016	1.19888E-016	-4.44089E-016	20	3.39999
500	500	0.5	0.18	3.52804E-016	1.19888E-016	-4.44089E-016	20	3.23999
600	500	0.6	0.18	-2.83176E-017	3.90464E-016	-4.44089E-016	20	3.23333
600	500	0.6	0.2	4.91341E-018	-7.7045E-017	-4.44089E-016	20	3.59999
700	500	0.6	0.15	-1.05479E-016	8.19869E-017	-4.44089E-016	20	2.65714
700	500	0.65	0.15	0	0	-4.44089E-016	20	2.59999
750	600	0.65	0.15	2.72501E-016	9.39578E-017	-4.44089E-016	20	2.74666
750	600	0.65	0.1	1.54428E-016	8.71311E-017	-4.44089E-016	20	1.78666
800	600	0.75	0.1	1.72729E-017	2.37178E-016	-4.44089E-016	20	1.75
800	600	0.75	0.07	-3.02337E-016	-2.17979E-016	-4.44089E-016	20	1.325
1000	1000	0.75	0.1	-2.01683E-017	-3.36791E-016	-4.44089E-016	20	1.9
1000	1200	0.75	0.1	1.07531E-016	2.7491E-016	-4.44089E-016	20	1.84
1200	1000	0.75	0.1	1.48644E-016	2.23442E-018	-4.44089E-016	20	1.9

Tabla 69: Resultados para la función de Ackley utilizando XHC y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (700), el número de generaciones (500), la tasa de supervivencia (0,65) y la tasa de mutación (0,15). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1.4668E-017	-5.16524E-017	-4.44089E-016	20	2.77142
19	1.30103E-016	3.30249E-016	-4.44089E-016	20	2.79999
43	2.41515E-016	2.7099E-016	-4.44089E-016	20	2.77142
83	8.80841E-017	-3.18375E-016	-4.44089E-016	20	2.68571
101	1.24109E-016	-4.1565E-017	-4.44089E-016	20	2.77142
137	3.33218E-016	7.49455E-017	-4.44089E-016	20	2.68571
151	0	0	-4.44089E-016	20	2.82856
191	-5.08498E-017	6.90219E-018	-4.44089E-016	20	2.68571
211	-1.42008E-016	-1.65786E-016	-4.44089E-016	20	2.82856
233	-1.02997E-016	2.67897E-016	-4.44089E-016	20	2.62857
277	-2.73075E-016	-1.11463E-016	-4.44089E-016	20	2.85714
313	-1.31603E-016	-1.88935E-016	-4.44089E-016	20	2.68571
349	0	0	-4.44089E-016	20	2.79999
467	1.24934E-016	-1.90014E-016	-4.44089E-016	20	2.62857
521	3.52808E-018	-1.87314E-017	-4.44089E-016	20	2.71428
599	-8.9354E-017	-3.39579E-016	-4.44089E-016	20	2.71428
617	-1.34386E-016	7.55642E-017	-4.44089E-016	20	2.71428
661	6.28384E-017	1.60618E-017	-4.44089E-016	20	2.79999
683	2.14403E-016	-2.02309E-016	-4.44089E-016	20	2.85714
701	2.96082E-016	1.12518E-016	-4.44089E-016	20	2.82856
727	-1.59446E-016	3.32094E-016	-4.44089E-016	20	2.82856
743	-1.89974E-016	-2.88283E-016	-4.44089E-016	20	2.91428
797	3.79088E-016	-1.04345E-017	-4.44089E-016	20	2.65714
811	1.50192E-016	-2.95978E-016	-4.44089E-016	20	2.82856
853	0	0	-4.44089E-016	20	2.77142
883	3.46115E-016	1.78311E-016	-4.44089E-016	20	2.68571
907	-1.14257E-016	-2.23145E-016	-4.44089E-016	20	2.79999
991	-3.29106E-016	9.58666E-017	-4.44089E-016	20	2.77142

Tabla 70: Resultados para la función de Ackley utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido	-4.44089E-016
Media de los resultados	-4.44089E-016
Desviación estándar de los resultados	0

**B.2.6. Esfera & GPAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.158456	0.719321	0.54253	221.805	18.5953
50	50	0.8	0.1	-0.267552	0.0870211	0.0791565	206.097	17.3871
100	50	0.8	0.1	-0.0544613	-0.181056	0.0357474	733.822	67.4059
150	50	0.8	0.1	0.206425	-0.053657	0.0454905	411.592	84.2509
150	150	0.8	0.1	0.0470242	-0.02212	0.00270057	599.367	84.9477
150	150	0.7	0.15	-0.0123825	-0.0522091	0.00287911	847.838	140.312
150	300	0.7	0.15	-0.0123825	-0.0522091	0.00287911	1184.97	150.925
250	300	0.7	0.15	0.0785378	-0.0216986	0.00663901	1974.02	238.819
350	300	0.7	0.15	-0.000980549	-0.0249339	0.000622659	2060.91	298.467
400	300	0.7	0.15	0.0475854	-0.0156431	0.00250908	4358.23	409.661
450	300	0.7	0.15	-0.0383368	-0.0609766	0.00518785	3117.19	392.918
500	300	0.7	0.15	-0.00250145	0.012562	0.000164061	2212.08	273.409
500	400	0.7	0.15	-0.00250145	0.012562	0.000164061	6981.91	619.283
500	400	0.6	0.15	-0.033116	0.013188	0.0012706	60460.7	4736.07
500	400	0.5	0.15	0.00447034	-0.0124508	0.000175007	460555	134156
500	400	0.5	0.18	0.0198851	-0.074071	0.00588193	310731	64293.1
500	500	0.5	0.18	0.0198851	-0.074071	0.00588193	438758	99630.5
600	500	0.6	0.18	-0.0553904	-0.0440722	0.00501046	47684.4	5198.74
600	500	0.6	0.2	-0.058142	0.0114269	0.00351106	101921	15067.9
700	500	0.6	0.15	-0.0221734	0.0121397	0.000639032	314262	63783
700	500	0.65	0.15	-0.0686477	0.0328475	0.00579147	45327	3725.5
750	600	0.65	0.15	0	0	0	33339	2812.47
750	600	0.65	0.1	0	0	0	13623.2	1742.27
800	600	0.75	0.1	0.0364643	0.010095	0.00143156	2596.94	257.468
800	600	0.75	0.07	-0.00866552	0.00785519	0.000136795	2087.43	219.917
1000	1000	0.75	0.1	-0.028592	-0.0287684	0.00164512	5996.69	518.209
1000	1200	0.75	0.1	-0.028592	-0.0287684	0.00164512	3226.4	378.529
1200	1000	0.75	0.1	-0.0176114	0.0198576	0.000704485	4266.4	422.085

Tabla 71: Resultados para la función Esfera utilizando GPAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0	0	0	23680.4	3347.94
19	0	0	0	17658.2	1795.21
43	-0.0217444	-0.0701621	0.00539553	28221.5	3337.83
83	0	0	0	18309.2	1307.49
101	-0.0137148	0.0135037	0.000370447	25375.5	3303.51
137	0	0	0	24010	2945.21
151	0	0	0	15352	2100.59
191	0	0	0	69094	5792.46
211	0	0	0	22226.3	3966.06
233	0	0	0	17174.7	1260.32
277	0	0	0	18588.2	2818.77
313	0	0	0	36850	4686.75
349	0	0	0	33236.2	3871.11
467	0	0	0	18174.1	2215.36
521	0	0	0	57003.6	3356.14
599	0	0	0	23723.8	3434.76
617	0	0	0	34778.2	4374.25
661	0	0	0	28766.9	2433.95
683	0	0	0	17747.7	1709.46
701	0	0	0	27202.9	3764.72
727	0	0	0	10207.7	1423.19
743	0	0	0	27812.7	2555.42
797	0	0	0	25705.2	1637.73
811	0	0	0	19150.7	1673.39
853	0	0	0	44703.5	5538.52
883	0	0	0	20329.1	3109.4
907	0	0	0	27419.7	5236.94
991	0	0	0	41142.4	3939.35

Tabla 72: Resultados para la función Esfera utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.0002059278
Desviación estándar de los resultados	0.0010194697

**B.2.7. Esfera & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.424707	0.136916	0.199122	219.701	11.3405
50	50	0.8	0.1	-0.0261558	-0.0397594	0.00226494	205.289	16.1004
100	50	0.8	0.1	-0.0321661	-0.00414344	0.00105182	534.49	21.5721
150	50	0.8	0.1	0.000861182	-0.0107403	0.000116096	918.819	71.7304
150	150	0.8	0.1	0.00821844	-0.00413911	8.46749E-005	946.61	82.3001
150	150	0.7	0.15	0.115416	0.0475855	0.0155852	1484.65	216.306
150	300	0.7	0.15	0.115416	0.0475855	0.0155852	1162.82	178.255
250	300	0.7	0.15	-0.0225002	0.0319289	0.00152571	1835.27	355.202
350	300	0.7	0.15	0.0324704	-0.0055834	0.0010855	3765.71	494.665
400	300	0.7	0.15	-0.0597205	0.0683807	0.00824247	3088.35	536.175
450	300	0.7	0.15	0.0679546	-0.0636052	0.00866345	3660.45	623.908
500	300	0.7	0.15	-0.00203639	0.000877565	0.000004917	4171.07	663.605
500	400	0.7	0.15	-0.00203639	0.000877565	0.000004917	3256.3	549.136
500	400	0.6	0.15	0.0699686	-0.0472082	0.00712422	3395.84	822.721
500	400	0.5	0.15	0.0429753	-0.0183634	0.00218409	6043.84	621.197
500	400	0.5	0.18	0.0729342	0.0936996	0.014099	4068.97	891.793
500	500	0.5	0.18	0.0729342	0.0936996	0.014099	4425.5	984.429
600	500	0.6	0.18	0.0464304	0.0517496	0.00483381	6962	1099.11
600	500	0.6	0.2	0.0882813	-0.0751013	0.0134338	5599.7	1440.46
700	500	0.6	0.15	0.00720017	0.0392594	0.00159314	4657.58	864.86
700	500	0.65	0.15	0	0	0	4059.63	790.367
750	600	0.65	0.15	0	0	0	5242.88	1055.7
750	600	0.65	0.1	0	0	0	3491.5	438.841
800	600	0.75	0.1	0.00222912	0.0112607	0.000131772	2292.16	329.522
800	600	0.75	0.07	-0.0146831	0.0163241	0.000482071	1455.64	164.813
1000	1000	0.75	0.1	-0.00801316	0.0186357	0.000411501	3133.12	407.951
1000	1200	0.75	0.1	-0.00801316	0.0186357	0.000411501	2985.64	361.487
1200	1000	0.75	0.1	-0.0239071	0.0543266	0.00352293	3331.28	519.218

Tabla 73: Resultados para la función Esfera utilizando Recombinación intermedia y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.00039851	0.0651643	0.00424655	2239.58	365.808
19	0	0	0	2860.69	385.195
43	0	0	0	2386.35	381.221
83	0	0	0	2657.32	426.913
101	-0.00480893	0.00407723	3.97496E-005	2447.63	478.574
137	0	0	0	2386.43	407.549
151	0.00560532	0.00862983	0.000105894	2192.1	383.313
191	0	0	0	2846.89	412.95
211	0	0	0	2893.97	360.465
233	0	0	0	2749.17	313.355
277	0	0	0	2513.09	421.476
313	0	0	0	2186.78	415.411
349	0.0397918	-0.0386542	0.00307754	2236.34	396.594
467	0	0	0	1900.6	340.865
521	0	0	0	3233.86	430.637
599	0	0	0	2707.97	405.829
617	0	0	0	2629.14	355.184
661	0	0	0	3605.91	431.945
683	0	0	0	2157.3	388.034
701	0	0	0	2709.39	437.611
727	0	0	0	2849.37	379.97
743	0	0	0	2757.29	392.218
797	0	0	0	3130.18	514.394
811	-0.00185082	0.0084539	0.000074894	3026.83	371.36
853	0	0	0	2577.6	392.095
883	0	0	0	2053.38	375.791
907	0	0	0	2386.12	382.533
991	0	0	0	2301.85	454.092

Tabla 74: Resultados para la función Esfera utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.000269451
Desviación estándar de los resultados	0.0009716335

**B.2.8. Esfera & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.528041	0.229557	0.331524	221.805	13.1379
50	50	0.8	0.1	0.000339452	-0.000247124	1.76297E-007	223.779	17.8116
100	50	0.8	0.1	-0.0142186	0.0250031	0.000827322	855.881	71.8577
150	50	0.8	0.1	-0.0855262	0.0507146	0.0098867	498.996	76.0419
150	150	0.8	0.1	0.00660593	-0.00647939	8.56208E-005	2591.11	104.821
150	150	0.7	0.15	0.0108285	-0.0149627	0.000341139	941.649	150.19
150	300	0.7	0.15	0.0122562	-0.0107898	0.000266634	1180.26	197.918
250	300	0.7	0.15	0.000143973	-0.00654903	4.29105E-005	2294.36	318.717
350	300	0.7	0.15	-0.00827271	0.00388385	8.35221E-005	3125.07	457.568
400	300	0.7	0.15	0.0134124	-0.0215183	0.000642931	3577.4	383.697
450	300	0.7	0.15	0.00555656	-0.000687787	3.13484E-005	2987.2	497.469
500	300	0.7	0.15	0.0161141	0.00853125	0.000332447	2421.74	390.684
500	400	0.7	0.15	0.0161141	0.00853125	0.000332447	4047.15	581.472
500	400	0.6	0.15	0.00781099	-0.00538246	8.99824E-005	4814.88	478.403
500	400	0.5	0.15	-0.0165366	0.00445629	0.000293318	7379.69	554.295
500	400	0.5	0.18	-0.000553573	0.000142415	3.26725E-007	4888.37	740.873
500	500	0.5	0.18	-0.000553573	0.000142415	3.26725E-007	4952.89	719.669
600	500	0.6	0.18	-0.000254361	0.00057971	4.00763E-007	7009.31	1127.35
600	500	0.6	0.2	0.0169052	-0.00510524	0.00031185	57067.6	1363.08
700	500	0.6	0.15	-0.0123989	-0.000852842	0.000154461	5425.29	785.142
700	500	0.65	0.15	0	0	0	5447.55	696.406
750	600	0.65	0.15	0	0	0	9590.77	687.604
750	600	0.65	0.1	0	0	0	2311.69	278.774
800	600	0.75	0.1	0.00628604	-0.00795819	0.000102847	4543.1	281.58
800	600	0.75	0.07	-0.00106538	0.00110983	2.36674E-006	15246.7	201.56
1000	1000	0.75	0.1	-0.00741194	0.00177222	5.80777E-005	5915.6	387.851
1000	1200	0.75	0.1	-0.00539246	0.00274073	3.65902E-005	3364.58	301.681
1200	1000	0.75	0.1	0.0106188	-0.0064003	0.000153723	3869.3	405.204

Tabla 75: Resultados para la función Esfera utilizando SBX y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.00186915	0.00527716	3.13421E-005	6202.1	445.671
19	0	0	0	5242.88	390.178
43	0	0	0	5320.4	447.396
83	0	0	0	3946.12	355.987
101	-0.0156381	-0.0104212	0.000353151	4936.47	360.191
137	0	0	0	3282.72	396.896
151	-0.00355616	0.000292241	1.27317E-005	3606.63	330.023
191	0	0	0	3355.44	376.075
211	0	0	0	3308.86	337.15
233	0	0	0	3706.81	344.405
277	0	0	0	2765.8	294.279
313	0	0	0	3541.12	379.951
349	0.00377231	-0.00131417	1.59574E-005	12794.1	403.501
467	0	0	0	3562.83	286.522
521	0	0	0	2766.62	306.198
599	0	0	0	3498.46	404.198
617	0	0	0	7775.23	365.556
661	0	0	0	2783.3	270.215
683	0	0	0	3540.23	374.984
701	0	0	0	3103.45	404.652
727	0	0	0	3162.95	293.07
743	0	0	0	3329	299.026
797	0	0	0	3241.2	298.89
811	0.00446629	-0.00515173	4.64881E-005	4229.55	488.33
853	0	0	0	2995.5	315.96
883	0	0	0	12286.9	371.396
907	0	0	0	5437.09	391.601
991	0	0	0	3124.7	352.598

Tabla 76: Resultados para la función Esfera utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido 0  
 Media de los resultados 1.641680E-005  
 Desviación estándar de los resultados 0.000066869

**B.2.9. Esfera & SBX adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0122742	-0.00638287	0.000191397	202.135	11.0473
50	50	0.8	0.1	0.0482097	-0.344686	0.121133	223.217	17.9474
100	50	0.8	0.1	-0.00197157	0.00516321	3.05459E-005	265.652	24.7465
150	50	0.8	0.1	-0.00437696	0.0214397	0.000478821	829.348	90.4473
150	150	0.8	0.1	-5.53211E-005	0.000000315	3.06052E-009	838.859	20.2216
150	150	0.7	0.15	0.000251708	-0.00015325	8.68425E-008	905.234	119.663
150	300	0.7	0.15	0.000180618	-0.000222254	0.000000082	859.813	125.109
250	300	0.7	0.15	0.00153881	-0.0015254	4.69478E-006	1840.69	188.736
350	300	0.7	0.15	0.0132806	-0.0133198	0.000353791	1765.42	233.53
400	300	0.7	0.15	-0.000134924	0.000136073	3.67203E-008	1926.77	277.85
450	300	0.7	0.15	-0.00222913	2.46017E-005	4.96961E-006	3280.99	346.356
500	300	0.7	0.15	3.22389E-005	8.76036E-005	8.71373E-009	2311.65	421.565
500	400	0.7	0.15	3.41844E-005	-5.51009E-005	4.20468E-009	1904.5	288.414
500	400	0.6	0.15	-0.00212471	0.00121201	5.98335E-006	3154.38	377.411
500	400	0.5	0.15	-0.000719855	6.69764E-005	5.22677E-007	3458.1	294.147
500	400	0.5	0.18	0.00130508	-0.00106971	2.84751E-006	5573.45	576.987
500	500	0.5	0.18	0.00131901	-0.00102397	2.78831E-006	2311.41	402.9
600	500	0.6	0.18	0.000400542	-0.000474906	0.000000386	4276.03	684.581
600	500	0.6	0.2	-0.00418024	0.00217101	2.21877E-005	224926	1535.33
700	500	0.6	0.15	-0.000205208	0.0011933	1.46608E-006	3574.04	554.057
700	500	0.65	0.15	0	0	0	3344.69	401.845
750	600	0.65	0.15	0	0	0	3978.15	588.235
750	600	0.65	0.1	0	0	0	2198.26	191.905
800	600	0.75	0.1	0.000160539	-0.000127235	0.000000042	2799.97	189.349
800	600	0.75	0.07	-0.000016076	0.000021125	7.04701E-010	1780.11	57.3271
1000	1000	0.75	0.1	9.07689E-006	0.000012152	2.30062E-010	2073.03	206.836
1000	1200	0.75	0.1	9.07689E-006	0.000012152	2.30062E-010	2073.36	208.958
1200	1000	0.75	0.1	0.000462387	-0.000413236	3.84566E-007	3204.77	272.562

Tabla 77: Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	5.09062E-005	-2.42297E-005	3.17852E-009	1887.5	177.709
19	0	0	0	1875.22	182.893
43	0	0	0	1961	183.613
83	0	0	0	1879.63	185.27
101	7.46969E-006	-0.000048096	2.36902E-009	2038.75	174.039
137	0	0	0	1722.34	200.879
151	1.95089E-005	-2.54102E-005	0.000000001	1904.14	161.319
191	0	0	0	1791.67	174.593
211	0	0	0	1955.65	158.871
233	0	0	0	3286.02	199.141
277	0	0	0	997.466	155.525
313	0	0	0	1837.98	151.356
349	-0.000307544	0.000215623	1.41076E-007	1959.38	179.628
467	0	0	0	2041.91	207.447
521	0	0	0	1838.64	169.985
599	0	0	0	1982.98	185.049
617	0	0	0	1962.68	163.447
661	0	0	0	2038.31	182.866
683	0	0	0	2507.83	171.865
701	0	0	0	1997.82	206.94
727	0	0	0	1997.34	205.943
743	0	0	0	2107.12	207.927
797	0	0	0	2149.02	213.548
811	9.40473E-005	-7.03959E-005	1.38005E-008	1867.75	174.679
853	0	0	0	1970.74	169.457
883	0	0	0	1887.03	188.353
907	0	0	0	1978.45	181.716
991	0	0	0	1888.36	200.034

Tabla 78: Resultados para la función de Ackley utilizando SBX adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	5.766083E-009
Desviación estándar de los resultados	2.665189E-008

**B.2.10. Esfera & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.000327745	3.42636E-005	1.08591E-007	209.875	10.4992
50	50	0.8	0.1	-2.22801E-011	4.27083E-011	2.32041E-021	209.715	16.7772
100	50	0.8	0.1	-8.02333E-011	3.74544E-010	1.46721E-019	209.715	16.7772
150	50	0.8	0.1	-2.49237E-011	2.47785E-013	6.21251E-022	209.741	18.1755
150	150	0.8	0.1	4.39829E-031	-1.12988E-034	1.9345E-061	209.715	18.1753
150	150	0.7	0.15	6.9175E-029	-7.09605E-030	4.83553E-057	838.861	69.8368
150	300	0.7	0.15	1.20557E-049	-2.39249E-050	1.51063E-098	838.861	53.6247
250	300	0.7	0.15	2.55055E-048	1.94882E-048	1.03032E-095	838.861	78.4831
350	300	0.7	0.15	1.1815E-043	2.64391E-044	1.46583E-086	838.861	76.3233
400	300	0.7	0.15	-1.55178E-042	8.67465E-044	2.41553E-084	838.861	65.1845
450	300	0.7	0.15	9.82317E-045	8.96717E-046	9.72989E-089	838.861	71.1406
500	300	0.7	0.15	-6.89294E-042	9.68362E-042	1.41285E-082	1887.44	90.8954
500	400	0.7	0.15	5.46872E-044	-1.29499E-044	3.15839E-087	1229.77	70.3333
500	400	0.6	0.15	1.38787E-055	8.4211E-056	2.63534E-110	838.861	63.3188
500	400	0.5	0.15	-2.78724E-069	3.22606E-069	1.81761E-137	838.861	65.0283
500	400	0.5	0.18	1.30918E-054	-1.10959E-053	1.24832E-106	1887.44	75.1368
500	500	0.5	0.18	-2.87923E-055	3.8061E-055	2.27764E-109	1250.17	77.3308
600	500	0.6	0.18	6.58479E-042	4.13716E-042	6.04756E-083	1887.44	110.952
600	500	0.6	0.2	7.09532E-038	7.92012E-039	5.09708E-075	1887.44	128.806
700	500	0.6	0.15	8.62297E-050	-5.51181E-050	1.04736E-098	1887.44	87.4518
700	500	0.65	0.15	0	0	0	1887.44	105.537
750	600	0.65	0.15	0	0	0	1275.26	86.8098
750	600	0.65	0.1	0	0	0	1887.44	41.9299
800	600	0.75	0.1	5.85444E-073	9.60454E-072	9.259E-143	1224.34	63.0192
800	600	0.75	0.07	-2.25668E-083	2.6291E-083	1.20047E-165	838.861	32.2338
1000	1000	0.75	0.1	8.65695E-073	4.61886E-072	2.20833E-143	838.861	51.8844
1000	1200	0.75	0.1	6.56302E-074	1.80541E-074	4.63328E-147	986.29	71.6063
1200	1000	0.75	0.1	-1.48999E-064	1.23745E-064	3.75134E-128	1887.44	63.9466

Tabla 79: Resultados para la función Esfera utilizando XHC y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (700), el número de generaciones (500), la tasa de supervivencia (0,65) y la tasa de mutación (0,15). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0	0	0	1231.02	93.0254
19	0	0	0	1887.44	84.6808
43	0	0	0	988.067	87.6167
83	2.51196E-046	1.91182E-046	9.965E-092	1887.44	92.3664
101	3.3744E-042	2.03601E-042	1.55319E-083	1255.31	78.962
137	0	0	0	1170.25	81.8945
151	0	0	0	1887.44	105.149
191	0	0	0	2263.55	85.0303
211	0	0	0	1887.44	95.6894
233	0	0	0	1263.94	85.118
277	0	0	0	1252.17	88.7519
313	0	0	0	1887.44	96.1132
349	0	0	0	2431.15	94.0005
467	0	0	0	1887.44	92.8613
521	0	0	0	1887.44	97.1387
599	0	0	0	2085.36	88.0275
617	3.9854E-044	1.15327E-043	1.48887E-086	843.258	82.4566
661	0	0	0	1887.44	94.3041
683	0	0	0	1887.44	98.1848
701	-2.34461E-044	2.72481E-044	1.29218E-087	1887.44	89.0211
727	5.62555E-046	-1.2764E-045	1.94566E-090	1887.44	81.0116
743	0	0	0	1887.44	98.7357
797	0	0	0	1887.44	103.847
811	-1.60225E-044	8.55095E-045	3.29841E-088	1887.44	90.0206
853	0	0	0	2049.14	104.166
883	0	0	0	1887.44	94.3832
907	0	0	0	1887.44	87.8054
991	0	0	0	1249.96	90.7802

Tabla 80: Resultados para la función Esfera utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido 0  
 Media de los resultados 5.553005E-085  
 Desviación estándar de los resultados 2.935139E-084

**B.2.11. Griewank & GPAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-2.97685	4.90548	0.0740777	703.039	36.0249
50	50	0.8	0.1	-3.05617	-4.6125	0.0184572	711.154	43.4717
100	50	0.8	0.1	-0.120172	0.0947118	0.00944336	720.071	64.1416
150	50	0.8	0.1	-3.05617	-4.6125	0.0184572	697.195	70.5275
150	150	0.8	0.1	-3.05617	-4.6125	0.0184572	1198.75	190.083
150	150	0.7	0.15	-3.05617	-4.6125	0.0184572	3529.79	506.088
150	300	0.7	0.15	-3.05617	-4.6125	0.0184572	3908.89	567.558
250	300	0.7	0.15	3.19743	-4.60107	0.0156467	7254.49	831.816
350	300	0.7	0.15	0.0180547	0.127842	0.00424959	8565.44	884.185
400	300	0.7	0.15	0.0145988	0.0905578	0.00215792	15188.7	1315.64
450	300	0.7	0.15	-3.12031	-4.56888	0.0118445	17468.6	1582.43
500	300	0.7	0.15	0.075924	0.180693	0.0110183	11915.8	1470.1
500	400	0.7	0.15	0.0728446	0.127774	0.00672534	13545.2	1215.16
500	400	0.6	0.15	0.00811747	-0.0963438	0.00235484	46868.8	5518.38
500	400	0.5	0.15	-3.16424	-4.56906	0.0119555	4812610	1628670
500	400	0.5	0.18	-3.16424	-4.56906	0.0119555	1475120	369110
500	500	0.5	0.18	-3.16424	-4.56906	0.0119555	2909600	580126
600	500	0.6	0.18	0.145346	-0.00465642	0.0105548	206839	35474.9
600	500	0.6	0.2	-0.068068	-0.165884	0.00917933	500045	62240.8
700	500	0.6	0.15	-6.2806	0.0114353	0.00989755	526630	90221.2
700	500	0.65	0.15	-3.16424	-4.56906	0.0119555	55540.3	6657.54
750	600	0.65	0.15	0	0	0	122366	13244
750	600	0.65	0.1	0	0	0	65626.9	8193.6
800	600	0.75	0.1	-0.04087	-0.154529	0.00680036	7883.61	896.169
800	600	0.75	0.07	-0.0470953	0.00552997	0.00111698	4738.02	544.585
1000	1000	0.75	0.1	0.0342017	-0.0861322	0.00244	13510.8	1440.15
1000	1200	0.75	0.1	0.0342017	-0.0861322	0.00244	13581.1	1388.22
1200	1000	0.75	0.1	3.10551	-4.47332	0.00829604	21321.2	1428.85

Tabla 81: Resultados para la función de Griewank utilizando GPAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0	0	0	132593	11924
19	0	0	0	64893.5	8863.54
43	0.0833639	-0.0649825	0.00452739	166610	15668.6
83	0	0	0	63264.8	5036.76
101	0.0837153	0.00187937	0.00350471	90890.6	10749.6
137	0	0	0	95861.4	12834.3
151	0	0	0	64962.3	8038.9
191	0	0	0	171350	24455.7
211	0	0	0	80753.3	8494.43
233	0	0	0	115581	12155.6
277	0	0	0	74194.8	9192.38
313	0	0	0	82226.8	10465.6
349	0	0	0	59111.3	8167.78
467	0	0	0	71382.5	10215.1
521	0	0	0	94175.3	10193.5
599	0	0	0	113224	10721.9
617	0	0	0	104411	11852.8
661	0	0	0	130098	15673.3
683	0	0	0	76251.6	6376.17
701	0	0	0	118612	14508.2
727	0	0	0	69137.1	6205.39
743	0	0	0	92340.7	9595.22
797	0	0	0	105427	12531
811	0	0	0	80250.1	10153.2
853	0	0	0	79984.4	6528.05
883	0	0	0	127014	14921.4
907	0	0	0	109199	10331.5
991	0	0	0	61092.5	9052.41

Tabla 82: Resultados para la función de Griewank utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.00028686
Desviación estándar de los resultados	0.00106242

**B.2.12. Griewank & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-25.2387	35.3192	0.489154	721.921	37.9033
50	50	0.8	0.1	-3.06513	-4.65931	0.022351	730.227	58.6222
100	50	0.8	0.1	3.09495	-3.27298	0.328902	2919.63	101.77
150	50	0.8	0.1	3.18939	12.9042	0.0899749	3009.73	228.209
150	150	0.8	0.1	3.25752	12.9723	0.0828084	3077.96	214.994
150	150	0.7	0.15	37.7964	8.7056	0.388878	7706.06	1022.65
150	300	0.7	0.15	37.7964	8.7056	0.388878	4239.89	682.819
250	300	0.7	0.15	-12.2562	-0.209152	0.0956651	6781.23	1099.7
350	300	0.7	0.15	-15.4955	3.97646	0.139145	10064.5	2020.93
400	300	0.7	0.15	-5.85516	0.0974381	0.100943	11055.6	1808.14
450	300	0.7	0.15	3.4344	-4.06247	0.0840668	13530.2	1850.09
500	300	0.7	0.15	-9.46803	22.3214	0.150762	13095.3	2483.21
500	400	0.7	0.15	-9.46803	22.3214	0.150762	11432.6	1989.28
500	400	0.6	0.15	31.8578	0.384735	0.383067	13814.1	2577.57
500	400	0.5	0.15	0.178391	9.17638	0.0576352	11595.5	2172.32
500	400	0.5	0.18	16.0434	13.8299	0.226589	16497.7	3480.88
500	500	0.5	0.18	16.0434	13.8299	0.226589	16751.6	3155.73
600	500	0.6	0.18	22.294	4.74201	0.196662	20052.1	4364.87
600	500	0.6	0.2	18.8808	9.06557	0.118224	19778.7	4706.01
700	500	0.6	0.15	-15.8242	-3.82063	0.167604	16499.3	2589.52
700	500	0.65	0.15	0	0	0	14845.7	3361.55
750	600	0.65	0.15	0	0	0	16651.3	3078.34
750	600	0.65	0.1	0	0	0	9824.25	1245.3
800	600	0.75	0.1	-18.6933	8.7643	0.12239	7389.42	1168.3
800	600	0.75	0.07	-12.571	17.77	0.118463	8971.25	638.133
1000	1000	0.75	0.1	13.0173	8.1185	0.288053	12597.3	1420.58
1000	1200	0.75	0.1	13.0173	8.1185	0.288053	11080.3	1483.58
1200	1000	0.75	0.1	-21.966	4.71805	0.145371	10119.5	1754.85

Tabla 83: Resultados para la función de Griewank utilizando Recombinación intermedia y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	12.5127	17.7539	0.119457	14013.9	1601.45
19	0	0	0	9917.62	1360.36
43	0	0	0	10971.4	1560.88
83	0	0	0	7719.41	1482.24
101	0.182164	-0.106101	0.0193236	6836.59	1396.39
137	0	0	0	7565.03	1509.22
151	6.21108	8.77623	0.0344884	6975.6	1672.15
191	0	0	0	7113.41	1225.93
211	0	0	0	11322.2	1710.59
233	0	0	0	10771.3	1175.87
277	0	0	0	7678.12	1427.12
313	0	0	0	8222.43	1386.4
349	12.5751	9.07115	0.0687214	7326.45	1222.65
467	0	0	0	7389.19	1245.44
521	0	0	0	9504.7	1368.83
599	0	0	0	8385.85	1566.54
617	0	0	0	7506.93	1290.47
661	0	0	0	11289.8	1488.64
683	0	0	0	8224.03	1264.44
701	0	0	0	7772.43	1197.01
727	0	0	0	7766.28	1396.46
743	0	0	0	10364.6	1378.43
797	0	0	0	8939.92	1656.52
811	3.30723	-4.25531	0.0296108	8613.16	1102.39
853	0	0	0	9549.26	1316.19
883	0	0	0	9430.01	1115.46
907	0	0	0	8653.81	1254.83
991	0	0	0	7867.17	1311.48

Tabla 84: Resultados para la función de Griewank utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.00970004
Desviación estándar de los resultados	0.02638582

**B.2.13. Griewank & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-13.4917	-8.22313	0.525673	763.398	43.8579
50	50	0.8	0.1	-9.47028	3.87086	0.107813	749.207	60.0656
100	50	0.8	0.1	6.19464	8.92733	0.0338657	1119.06	103.289
150	50	0.8	0.1	-5.92371	17.7646	0.151596	2601.37	278.436
150	150	0.8	0.1	3.32486	4.22234	0.0358987	3492.67	245.89
150	150	0.7	0.15	5.78973	8.93578	0.148193	4257.59	652.056
150	300	0.7	0.15	6.20149	8.94138	0.0337072	4422.14	593.58
250	300	0.7	0.15	6.42308	-9.10474	0.0526544	6740.33	1144.84
350	300	0.7	0.15	-0.205954	0.105217	0.023855	10501.5	1260.3
400	300	0.7	0.15	12.5808	0.0674249	0.0408096	17565.7	1308.79
450	300	0.7	0.15	-0.168653	17.7952	0.0935016	10389.9	1580.29
500	300	0.7	0.15	-3.30895	4.68957	0.0371708	13315.5	1795.8
500	400	0.7	0.15	-3.30895	4.68957	0.0371708	13458.1	1512.28
500	400	0.6	0.15	-18.7392	-8.4977	0.149112	14870.5	1629.9
500	400	0.5	0.15	-3.51705	4.42501	0.0777204	15889.1	1848.37
500	400	0.5	0.18	-15.8701	5.0392	0.168871	31235.3	2091.41
500	500	0.5	0.18	-15.8701	5.0392	0.168871	16220.1	2406.78
600	500	0.6	0.18	-9.83581	-4.09596	0.139115	21650.3	2915.44
600	500	0.6	0.2	0.169582	-0.0879893	0.016261	38838.8	4651.79
700	500	0.6	0.15	6.20715	-0.00587929	0.0125298	24761.7	3107.46
700	500	0.65	0.15	0	0	0	13428.8	2151.03
750	600	0.65	0.15	0	0	0	16254.1	2571.6
750	600	0.65	0.1	0	0	0	13779.2	1031.98
800	600	0.75	0.1	-6.40184	-8.73135	0.0422498	16952.4	963.025
800	600	0.75	0.07	3.07519	-4.38358	0.0102489	15149	677.502
1000	1000	0.75	0.1	6.35678	9.12897	0.0483546	10080.6	1019.04
1000	1200	0.75	0.1	6.15057	8.85793	0.0380453	12358.1	1126.32
1200	1000	0.75	0.1	-0.00302948	8.58714	0.0406515	10788.2	1078.04

Tabla 85: Resultados para la función de Griewank utilizando SBX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	6.09109	8.86032	0.0474546	7333.46	928.324
19	0	0	0	21714.3	1239.57
43	0	0	0	11520.7	1316.72
83	0	0	0	9877.08	1021.12
101	-12.5819	-9.12358	0.0746104	17828.7	1457.78
137	0	0	0	9397.25	1039.75
151	-15.5969	-4.46409	0.0720674	23057.8	1188.95
191	0	0	0	26581.1	1595.02
211	0	0	0	17649.3	1090.41
233	0	0	0	12658.5	1213.25
277	0	0	0	32088.2	1378.18
313	0	0	0	13206.1	1235.4
349	9.20894	-4.89055	0.0989154	9287.37	1043.95
467	0	0	0	9723.8	1188.73
521	0	0	0	7581.62	972.183
599	0	0	0	21164.5	1144.24
617	0	0	0	33791.3	1175.88
661	0	0	0	9780.74	998.587
683	0	0	0	11488.8	1181.99
701	0	0	0	10212.2	1031.91
727	0	0	0	11508.1	1194.54
743	0	0	0	11964.6	1009.36
797	0	0	0	10793.9	1266.94
811	9.46984	13.2628	0.068493	30659.9	1643.86
853	0	0	0	15788.9	1529.37
883	0	0	0	62467.5	1454.19
907	0	0	0	54637.6	1235.49
991	0	0	0	12265.1	1350.76

Tabla 86: Resultados para la función de Griewank utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.01291217
Desviación estándar de los resultados	0.02907183

**B.2.14. Griewank & SBX adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-2.98576	4.86025	0.0629608	721.306	36.3479
50	50	0.8	0.1	-3.09924	-4.51282	0.0096111	715.516	57.3694
100	50	0.8	0.1	0.0137347	-0.0256572	0.000259084	724.654	58.3209
150	50	0.8	0.1	0.0218236	-0.010078	0.000263655	2873.95	210.521
150	150	0.8	0.1	-0.000192213	0.00019944	2.84363E-008	2805.03	71.8007
150	150	0.7	0.15	0.0340726	-0.0824489	0.00228039	2880.78	370.635
150	300	0.7	0.15	0.0367624	-0.0733953	0.00202285	2894.61	437.34
250	300	0.7	0.15	3.1572	-4.47485	0.00787524	3765.71	694.779
350	300	0.7	0.15	0.00435107	-0.00492346	1.55367E-005	11450.3	1083.14
400	300	0.7	0.15	-0.00434487	0.0117571	4.40349E-005	9349.18	1163.48
450	300	0.7	0.15	-3.1159	-4.48	0.00811897	7975.58	1545.47
500	300	0.7	0.15	0.0356887	-0.081907	0.00231442	12174.4	1883.9
500	400	0.7	0.15	0.0356887	-0.081907	0.00231442	11739.1	1523.11
500	400	0.6	0.15	0.015901	-0.0226708	0.000255082	9770.6	1397.2
500	400	0.5	0.15	-0.0150507	0.0333481	0.000391574	10772.5	872.455
500	400	0.5	0.18	0.00123494	0.0213497	0.000114827	13482	1948.5
500	500	0.5	0.18	0.00123494	0.0213497	0.000114827	12390.5	1739.46
600	500	0.6	0.18	-3.1351	-4.43361	0.00741402	19088.2	2276.48
600	500	0.6	0.2	0.0230603	-0.0977045	0.00265336	191654	3932.27
700	500	0.6	0.15	0.0298721	-0.0438933	0.000928245	15273.3	1712.03
700	500	0.65	0.15	0	0	0	12391.9	1758.58
750	600	0.65	0.15	0	0	0	13856.9	2178.79
750	600	0.65	0.1	0	0	0	6353.52	639.182
800	600	0.75	0.1	-0.00184638	0.00783032	1.70491E-005	6548.72	519.259
800	600	0.75	0.07	-0.000594022	0.000667767	2.88109E-007	2939.1	227.086
1000	1000	0.75	0.1	-0.001204	0.00120715	1.08984E-006	6914.69	722.076
1000	1200	0.75	0.1	-0.001204	0.00120715	1.08984E-006	6661.97	676.419
1200	1000	0.75	0.1	-0.000721808	0.00131095	6.9071E-007	6894.77	760.2
2								

Tabla 87: Resultados para la función de Griewank utilizando SBX adaptativo y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,1). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.000426278	-0.00128125	5.01711E-007	6452.77	627.307
19	0	0	0	6490.38	660.965
43	0	0	0	6121.47	627.997
83	0	0	0	7286.98	637.53
101	0.00416534	-0.0100385	3.38969E-005	6486.41	697.256
137	0	0	0	6202.39	642.362
151	-0.00927079	0.017615	0.00012064	6347.14	675.39
191	0	0	0	6161.12	570.679
211	0	0	0	6710.02	606.189
233	0	0	0	7086.66	571.884
277	0	0	0	7042.98	583.435
313	0	0	0	6645.58	617.137
349	0.00242075	-0.00388096	6.70069E-006	6541.92	634.911
467	0	0	0	6460.6	581.627
521	0	0	0	5867.8	646.08
599	0	0	0	6480.4	635.681
617	0	0	0	7547.64	792.307
661	0	0	0	6608.61	588.295
683	0	0	0	6441.73	646.738
701	0	0	0	6480.4	584.504
727	0	0	0	7131.51	776.427
743	0	0	0	7006.03	607.844
797	0	0	0	7367.34	584.173
811	-0.000421767	0.000868718	2.77845E-007	6752.93	619.262
853	0	0	0	6888.28	638.5
883	0	0	0	6599.73	618.291
907	0	0	0	6496.07	650.299
991	0	0	0	6619.44	621.906

Tabla 88: Resultados para la función de Griewank utilizando SBX adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido 0  
 Media de los resultados 5.786326E-006  
 Desviación estándar de los resultados 2.342106E-005

**B.2.15. Griewank & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-25.0282	-9.28775	0.223539	701.394	35.361
50	50	0.8	0.1	-3.14003	-4.43839	0.00739604	729.171	57.5231
100	50	0.8	0.1	-1.66178E-005	-0.00127116	4.04507E-007	720.847	59.1569
150	50	0.8	0.1	-0.000480793	0.000283981	1.3582E-007	727.711	62.5104
150	150	0.8	0.1	-2.45678E-010	-1.14481E-008	0	720.048	62.4041
150	150	0.7	0.15	0.000000007	-6.91552E-009	0	2880.19	199.839
150	300	0.7	0.15	6.69449E-009	-4.29282E-009	0	2880.19	194.512
250	300	0.7	0.15	-6.20236E-009	6.216E-009	0	2880.19	256.617
350	300	0.7	0.15	8.48816E-009	-1.23202E-009	0	2907.89	250.552
400	300	0.7	0.15	-6.74443E-009	6.17168E-009	0	2880.19	229.572
450	300	0.7	0.15	6.90094E-010	-6.60308E-009	0	2880.19	300.391
500	300	0.7	0.15	2.49186E-010	6.40649E-009	0	6480.4	312.636
500	400	0.7	0.15	8.85062E-009	-7.26399E-009	0	2880.19	296.635
500	400	0.6	0.15	-0.000000008	4.94624E-009	0	6480.4	270.011
500	400	0.5	0.15	7.45046E-009	1.80003E-009	0	2880.19	213.313
500	400	0.5	0.18	9.83458E-009	2.88544E-010	0	4149.58	272.903
500	500	0.5	0.18	-5.53765E-009	9.16516E-009	0	3417.57	257.471
600	500	0.6	0.18	3.19841E-009	1.74528E-010	0	6480.4	376.681
600	500	0.6	0.2	-1.47854E-009	-2.62615E-009	0	6590.93	435.688
700	500	0.6	0.15	1.89277E-009	1.43847E-008	0	6480.4	302.176
700	500	0.65	0.15	-1.07181E-009	-5.49579E-009	0	6480.4	345.257
750	600	0.65	0.15	5.72623E-009	6.51189E-009	0	6480.4	313.564
750	600	0.65	0.1	-1.63742E-009	7.08639E-009	0	6480.4	153.162
800	600	0.75	0.1	-2.10748E-009	-0.000000013	0	2880.19	178.103
800	600	0.75	0.07	-1.25455E-009	-8.37143E-009	0	2880.19	111.008
1000	1000	0.75	0.1	-5.26855E-009	1.14445E-008	0	2880.19	234.831
1000	1200	0.75	0.1	-9.17739E-009	3.23037E-010	0	3613.42	179.922
1200	1000	0.75	0.1	9.75591E-009	-4.16727E-009	0	2907.88	177.709

Tabla 89: Resultados para la función de Griewank utilizando XHC y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-7.21971E-011	-6.10493E-009	0	2880.19	108.558
19	4.89212E-009	9.15829E-009	0	2880.19	119.606
43	-3.2517E-009	-5.27947E-009	0	2880.19	123.545
83	-4.65451E-009	-0.000000012	0	2880.19	111.345
101	-2.88246E-009	-5.34336E-009	0	2880.19	113.563
137	-1.00753E-008	1.14126E-008	0	2880.19	117.007
151	-3.2584E-009	-3.80623E-009	0	2880.19	116.164
191	-3.73731E-009	4.71647E-009	0	2880.19	103.144
211	-2.32852E-009	-0.000000005	0	2880.19	114.249
233	-5.32973E-009	-1.46776E-008	0	2880.19	107.901
277	-1.00785E-008	1.05342E-008	0	2880.19	99.7243
313	6.64058E-009	-3.49729E-009	0	2880.19	120.057
349	-6.89802E-010	6.62061E-009	0	2880.19	122.533
467	-3.31253E-010	-1.37751E-008	0	2880.19	122.84
521	8.19087E-009	-1.26395E-009	0	3709.64	108.763
599	6.1981E-009	-3.62394E-009	0	2880.19	112.934
617	-1.0341E-008	-1.0667E-008	0	2880.19	125.231
661	-7.62391E-009	-1.10635E-008	0	2880.19	119.846
683	-7.11533E-009	-6.60517E-009	0	2880.19	110.485
701	-7.3811E-010	7.8327E-009	0	2880.19	111.194
727	5.64829E-009	-1.30187E-009	0	2880.19	113.734
743	-8.55977E-009	5.61274E-009	0	2880.19	118.588
797	-3.91471E-010	1.1682E-008	0	2880.2	122.516
811	-2.76513E-009	-1.2374E-008	0	2880.19	107.947
853	3.93253E-009	-1.45846E-008	0	2880.19	122.449
883	-9.00965E-011	-1.14852E-008	0	2880.19	107.167
907	-4.19036E-009	-2.71631E-009	0	2880.19	122.341
991	-0.000000006	0.000000006	0	2880.19	120.236

Tabla 90: Resultados para la función de Griewank utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0
Desviación estándar de los resultados	0

**B.2.16. Rastrigin & GPAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.99165	1.9858	4.98027	292.959	22.2276
50	50	0.8	0.1	-0.0368081	1.02093	1.39622	248.704	23.8686
100	50	0.8	0.1	-0.0368082	1.01947	1.38165	478.179	44.9115
150	50	0.8	0.1	-0.0370718	1.01878	1.37894	517.026	95.0053
150	150	0.8	0.1	-0.0167696	0.997007	1.05153	513.245	93.7448
150	150	0.7	0.15	1.01787	0.0231378	1.2051	1100.46	175.449
150	300	0.7	0.15	-0.0433203	0.0637779	1.16633	922.374	200.523
250	300	0.7	0.15	-0.032005	0.0186239	0.271273	1153.97	207.46
350	300	0.7	0.15	0.0366287	0.0140343	0.304059	3835.76	232.149
400	300	0.7	0.15	0.0247364	-0.0147198	0.164108	6903.03	405.561
450	300	0.7	0.15	0.0524841	-0.0367317	0.808076	3848.71	456.259
500	300	0.7	0.15	-0.0239616	1.02153	1.24858	2998.13	394.176
500	400	0.7	0.15	-0.0239616	1.02153	1.24858	3496.21	434.481
500	400	0.6	0.15	-0.0128562	-0.00763802	0.0443451	16298.8	2098.63
500	400	0.5	0.15	-0.0374234	1.01745	1.37188	2022500	693386
500	400	0.5	0.18	-0.00295632	-0.00629518	0.00959503	819025	252527
500	500	0.5	0.18	-0.00295632	-0.00629518	0.00959503	1358320	314409
600	500	0.6	0.18	-0.0454042	-0.0620131	1.15963	48266.1	5173.16
600	500	0.6	0.2	-0.0411159	-0.988747	1.33613	120693	8887.94
700	500	0.6	0.15	0.00959636	-0.0311535	0.210202	127053	27530.3
700	500	0.65	0.15	-0.992227	-0.00944684	1.01413	31825.9	3384.24
750	600	0.65	0.15	0	0	0	25492.9	3342.65
750	600	0.65	0.1	0	0	0	24013.4	2695.91
800	600	0.75	0.1	0.0356526	-0.0232519	0.358203	4020.59	289.66
800	600	0.75	0.07	-0.00578646	0.00141821	0.00704112	3658.44	225.816
1000	1000	0.75	0.1	-0.022556	0.00708957	0.110739	5603.04	586.42
1000	1200	0.75	0.1	-0.022556	0.00708957	0.110739	2865.9	265.64
1200	1000	0.75	0.1	0.00672856	0.0279756	0.163853	7490.94	568.721

Tabla 91: Resultados para la función de Rastrigin utilizando GPAX y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0	0	0	55135.2	2445.54
19	0	0	0	23733.6	2541.42
43	-0.0137642	-0.0197491	0.114843	24183.6	2658.32
83	0	0	0	17960.5	1950.37
101	-0.0470058	0.00524596	0.440657	46183.2	5220.26
137	0	0	0	34163.7	2463.05
151	0	0	0	30676.9	2639.53
191	0	0	0	26935.7	3205.55
211	0	0	0	23535.7	3845.45
233	0	0	0	24675.5	2607.04
277	0	0	0	43386.6	5592.33
313	0	0	0	50474.6	3996.73
349	0	0	0	33193.1	5415.2
467	0	0	0	17642.5	2473.93
521	0	0	0	23921.6	2811.48
599	0	0	0	13576.8	2334.23
617	0	0	0	19150.1	2911.83
661	0	0	0	27577.3	2847.37
683	0	0	0	24438.9	2728.26
701	0	0	0	24717.7	4046.46
727	0	0	0	25097.3	3209.75
743	0	0	0	25765.7	1938.84
797	0	0	0	34782.5	4817.61
811	0	0	0	21442.9	1713.73
853	0	0	0	21809.7	1552.45
883	0	0	0	37059.9	3938.74
907	0	0	0	38639	5765.82
991	0	0	0	26168.9	1616.58

Tabla 92: Resultados para la función de Rastrigin utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.01983928
Desviación estándar de los resultados	0.08527662

**B.2.17. Rastrigin & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0663276	0.122583	3.69772	292.959	20.9835
50	50	0.8	0.1	-5.62318E-005	1.00172	1.00404	248.793	22.1322
100	50	0.8	0.1	-0.00331342	0.00491593	0.00697209	441.859	37.0926
150	50	0.8	0.1	-0.005413	0.0131816	0.0402647	913.934	85.1965
150	150	0.8	0.1	-0.005413	0.0131816	0.0402647	1070.52	109.502
150	150	0.7	0.15	-0.0213085	1.01138	1.13842	1373.67	263.635
150	300	0.7	0.15	-0.0191999	1.01349	1.13612	998.797	218.674
250	300	0.7	0.15	0.0174469	1.01214	1.11386	2944.73	413.418
350	300	0.7	0.15	-0.0324961	1.02522	1.3852	3229.24	504.661
400	300	0.7	0.15	-0.0292182	1.02844	1.38586	2098.74	483.878
450	300	0.7	0.15	0.00478182	0.0308821	0.193154	3680.42	708.411
500	300	0.7	0.15	-0.0294257	1.02745	1.37538	4013.01	733.686
500	400	0.7	0.15	-0.0294257	1.02745	1.37538	3762.39	654.906
500	400	0.6	0.15	0.0288269	-0.0326706	0.375434	4362.14	775.379
500	400	0.5	0.15	0.0142518	-0.0208879	0.126706	6009.41	701.316
500	400	0.5	0.18	-0.0191363	1.01837	1.17619	4393.93	996.89
500	500	0.5	0.18	-0.0191363	1.01837	1.17619	3818.5	943.693
600	500	0.6	0.18	-0.0238142	0.0252868	0.238894	5853.48	1246.35
600	500	0.6	0.2	0.0423048	-0.0264064	0.491012	6193.13	1645.75
700	500	0.6	0.15	0.00823815	0.987405	1.01972	4638.3	967.221
700	500	0.65	0.15	0	0	0	4294.12	935.17
750	600	0.65	0.15	0	0	0	3506.92	849.976
750	600	0.65	0.1	0	0	0	2295.3	349.812
800	600	0.75	0.1	0.0152283	0.987748	1.05122	3038.01	384.658
800	600	0.75	0.07	-0.0256574	0.0355711	0.380311	1450.49	221.28
1000	1000	0.75	0.1	-0.00988785	0.0632885	0.803677	3023.26	445.996
1000	1200	0.75	0.1	-0.00988785	0.0632885	0.803677	2921.44	419.613
1200	1000	0.75	0.1	0.0601865	-0.00728576	0.720711	2725.46	505.622

Tabla 93: Resultados para la función de Rastrigin utilizando Recombinación intermedia y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.00374301	0.0283017	0.161274	2477.43	392.758
19	0	0	0	2652.82	470.022
43	0	0	0	2722.2	476.179
83	0	0	0	2576.75	489.565
101	-0.0286413	0.0345277	0.397905	1817.11	408.529
137	0	0	0	2285.59	396.571
151	0.0104532	0.0103855	0.0430611	2142.51	422.221
191	0	0	0	1949.81	351.053
211	0	0	0	3316.24	472.447
233	0	0	0	3215.91	471.438
277	0	0	0	2118.62	419.342
313	0	0	0	2722.01	404.278
349	-0.00968299	0.00966984	0.0371409	3014.33	404.194
467	0	0	0	2905.31	469.439
521	0	0	0	2154.44	480.042
599	0	0	0	2656.95	423.983
617	0	0	0	3024.3	423.197
661	0	0	0	3735.19	450.395
683	0	0	0	2704.58	416.805
701	0	0	0	2435.45	471.891
727	0	0	0	3357.33	439.957
743	0	0	0	2728.43	403.587
797	0	0	0	2434.72	524.147
811	-0.00487732	0.0117896	0.032282	2709.97	371.754
853	0	0	0	2812.88	390.652
883	0	0	0	2133.78	436.7
907	0	0	0	2720.67	395.951
991	0	0	0	2569.19	424.301

Tabla 94: Resultados para la función de Rastrigin utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0.02398796
Desviación estándar de los resultados	0.07993091

**B.2.18. Rastringin & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-1.0546	0.0562607	2.31647	206.457	22.5321
50	50	0.8	0.1	-0.00112423	0.00284146	0.00185251	236.734	23.9242
100	50	0.8	0.1	-1.00064	-0.00949904	1.01925	293.341	36.5721
150	50	0.8	0.1	-0.0248634	-1.00198	1.12713	690.764	84.1687
150	150	0.8	0.1	0.00229612	0.00399018	0.00420451	1088.47	94.4005
150	150	0.7	0.15	-1.0068	0.0143207	1.06344	1439.22	237.184
150	300	0.7	0.15	0.00721071	-0.00456457	0.0144469	2066.08	245.943
250	300	0.7	0.15	0.00173171	0.0472145	0.439637	2564.37	357.3
350	300	0.7	0.15	0.0134251	-0.0522874	0.5733	2929.58	396.724
400	300	0.7	0.15	-0.00420621	0.00339567	0.00579731	3273.09	429.293
450	300	0.7	0.15	0.00353974	0.990518	1.00135	2583.28	452.621
500	300	0.7	0.15	0.000109974	0.0109328	0.0237064	4611.77	615.679
500	400	0.7	0.15	-0.000233948	-0.00776484	0.0119701	3924.87	526.103
500	400	0.6	0.15	0.00498648	-0.00278146	0.00646746	3740.62	573.428
500	400	0.5	0.15	-0.019142	0.0182515	0.138624	4198.54	518.51
500	400	0.5	0.18	-0.00774198	0.00424587	0.0154653	5549.57	720.871
500	500	0.5	0.18	-0.00774198	0.00424587	0.0154653	4170.81	614.677
600	500	0.6	0.18	-0.0272276	0.0303132	0.328474	11834.8	1083.32
600	500	0.6	0.2	-0.00294229	-0.00517083	0.00702152	17611.7	1281.29
700	500	0.6	0.15	-0.00974707	0.00523517	0.0242794	6576.31	703.458
700	500	0.65	0.15	0	0	0	4313.93	838.779
750	600	0.65	0.15	0	0	0	6637.47	641.243
750	600	0.65	0.1	0	0	0	4126.05	379.932
800	600	0.75	0.1	0.00727016	0.00319665	0.0125115	3495.81	360.331
800	600	0.75	0.07	8.36376E-006	0.00102277	0.000207544	18291	179.703
1000	1000	0.75	0.1	0.000922161	-0.00140764	0.000561814	3163.94	320.691
1000	1200	0.75	0.1	0.000922161	-0.00140764	0.000561814	3066.28	294.982
1200	1000	0.75	0.1	-0.00081271	-0.002232	0.00111938	5354.36	544.209

Tabla 95: Resultados para la función de Rastringin utilizando SBX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-0.00685443	0.0153981	0.0563223	2613.6	299.433
19	0	0	0	5862.38	419.224
43	0	0	0	3354.22	359.331
83	0	0	0	6629.03	406.51
101	-0.00025624	0.000119722	1.58699E-005	8594.98	528.929
137	0	0	0	3042.84	348.23
151	0.0126873	-0.00565663	0.0382653	5507.56	340.154
191	0	0	0	2973.31	360.345
211	0	0	0	2473.02	342.383
233	0	0	0	3373.42	416.886
277	0	0	0	4123.6	291.722
313	0	0	0	5930.1	420.718
349	-0.0161178	-0.00402105	0.054703	3367.27	283.632
467	0	0	0	6021.31	407.475
521	0	0	0	2951.24	383.807
599	0	0	0	4603.65	431.809
617	0	0	0	3356.07	371.561
661	0	0	0	3852.09	399.686
683	0	0	0	4648.27	388.484
701	0	0	0	3165.97	430.512
727	0	0	0	5882.32	440.797
743	0	0	0	3997.67	342.795
797	0	0	0	3573.33	405.666
811	-0.00885265	0.00536042	0.0212441	8351.09	405.213
853	0	0	0	3000.67	327.761
883	0	0	0	72070.2	422.936
907	0	0	0	7053.82	414.51
991	0	0	0	3490.72	350.155

Tabla 96: Resultados para la función de Rastrigin utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido 0  
 Media de los resultados 0.00609109  
 Desviación estándar de los resultados 0.01614898

**B.2.19. Rastrigin & SBX Adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.993111	0.980742	2.03058	233.311	15.0978
50	50	0.8	0.1	-0.00243954	0.995576	0.996211	248.317	21.7256
100	50	0.8	0.1	-0.000527544	0.0141692	0.0398599	797.962	32.5802
150	50	0.8	0.1	0.00460542	0.00936587	0.0216055	966.726	83.9902
150	150	0.8	0.1	-0.000025639	2.00729E-005	2.10352E-007	881.656	65.7179
150	150	0.7	0.15	-0.000276587	0.994935	0.99497	927.035	136.775
150	300	0.7	0.15	-1.67831E-006	0.994978	0.994955	926.731	134.416
250	300	0.7	0.15	-0.000009157	6.1353E-006	2.41032E-008	1913.18	251.528
350	300	0.7	0.15	-1.84445E-005	0.995103	0.994959	1973.57	280.873
400	300	0.7	0.15	-6.39864E-005	0.00026277	0.000014511	3133.22	299.952
450	300	0.7	0.15	-0.000731304	-0.000904922	0.000268562	3188.55	351.14
500	300	0.7	0.15	0.000164619	-0.000112253	7.87624E-006	1945.41	285.109
500	400	0.7	0.15	0.00011387	-0.000110704	5.00384E-006	2064.26	277.087
500	400	0.6	0.15	-0.000845715	0.000678375	0.000233196	2040.77	376.136
500	400	0.5	0.15	3.54913E-005	1.04021E-005	2.71369E-007	2038.28	310.256
500	400	0.5	0.18	-7.71795E-005	0.0007214	0.000104429	3500.26	503.988
500	500	0.5	0.18	-0.000392674	-0.000158995	0.000035606	3291.84	405.009
600	500	0.6	0.18	4.8874E-006	-0.000243517	1.17696E-005	5379.46	688.496
600	500	0.6	0.2	0.00178417	-0.00124906	0.000941052	5185.61	799.352
700	500	0.6	0.15	-0.000115978	-3.27554E-005	2.88144E-006	3022.87	501.096
700	500	0.65	0.15	0	0	0	3232.85	492.543
750	600	0.65	0.15	0	0	0	3678.88	531.224
750	600	0.65	0.1	0	0	0	2034.94	200.695
800	600	0.75	0.1	1.12809E-005	-3.58815E-005	2.80674E-007	1908.88	171.74
800	600	0.75	0.07	-7.39766E-006	3.35053E-006	1.30843E-008	889.385	78.248
1000	1000	0.75	0.1	-2.14297E-005	4.80483E-005	5.49128E-007	2226.83	203.103
1000	1200	0.75	0.1	-4.14111E-005	2.66947E-005	4.81596E-007	1928.74	193.073
1200	1000	0.75	0.1	4.22755E-005	-3.48757E-005	5.9588E-007	3329.62	212.213

Tabla 97: Resultados para la función de Rastrigin utilizando SBX Adaptativo y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-2.34799E-005	4.19156E-006	1.1286E-007	2034.77	216.72
19	0	0	0	2024.35	208.816
43	0	0	0	2000.38	210.186
83	0	0	0	1904.08	177.82
101	-8.03045E-005	-0.000160908	6.41607E-006	1908.64	211.03
137	0	0	0	1961.54	216.374
151	-0.000189828	0.000140642	1.10733E-005	1968.04	211.574
191	0	0	0	1915.59	192.014
211	0	0	0	1914.02	203.987
233	0	0	0	1923.98	207.894
277	0	0	0	1845.84	195.737
313	0	0	0	2015.32	183.91
349	2.17154E-006	8.97973E-006	1.69331E-008	1911.09	185.081
467	0	0	0	2048.82	205.385
521	0	0	0	1804.85	206.038
599	0	0	0	3026.84	202.274
617	0	0	0	1911.18	194.87
661	0	0	0	2164.79	212.611
683	0	0	0	2147.33	216.678
701	0	0	0	1056.04	165.387
727	0	0	0	1940.54	228.354
743	0	0	0	2080.85	213.299
797	0	0	0	1976.62	221.684
811	0.000031552	-3.51968E-005	4.43278E-007	1916.11	192.769
853	0	0	0	1911.18	192.37
883	0	0	0	2095.28	210.829
907	0	0	0	2325.02	203.59
991	0	0	0	1671.5	187.81

Tabla 98: Resultados para la función de Rastrigin utilizando SBX Adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	6.450871E-007
Desviación estándar de los resultados	2.375348E-006

**B.2.20. Rastringin & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-1.00305	0.994741	2.00293	230.101	13.3187
50	50	0.8	0.1	-2.66326E-007	7.14154E-008	1.50848E-011	228.821	18.2951
100	50	0.8	0.1	-3.27349E-010	-3.00279E-011	0	228.464	18.277
150	50	0.8	0.1	-1.1662E-008	3.65337E-009	2.84217E-014	249.493	20.1601
150	150	0.8	0.1	-1.30435E-009	1.36058E-009	0	228.462	19.8001
150	150	0.7	0.15	-1.48565E-009	-4.78623E-010	0	878.704	62.1562
150	300	0.7	0.15	-4.93554E-010	8.85256E-010	0	878.704	57.3167
250	300	0.7	0.15	-1.28516E-009	1.08307E-009	0	878.704	70.7497
350	300	0.7	0.15	4.90568E-010	-4.80089E-010	0	878.704	93.9951
400	300	0.7	0.15	-0.000000001	-3.14586E-010	0	1334.27	93.148
450	300	0.7	0.15	1.49807E-009	1.10311E-009	0	1340.64	93.3182
500	300	0.7	0.15	-1.38351E-009	-9.78719E-011	0	1911.18	91.3651
500	400	0.7	0.15	-7.85363E-010	-4.7474E-010	0	878.704	80.3715
500	400	0.6	0.15	-0.000000001	-9.60704E-011	0	1911.18	80.203
500	400	0.5	0.15	-1.44498E-010	-1.32537E-009	0	878.704	63.8494
500	400	0.5	0.18	7.21559E-010	-8.72967E-010	0	1911.18	97.9551
500	500	0.5	0.18	-9.28304E-010	2.85119E-010	0	1049.95	77.1505
600	500	0.6	0.18	-1.19024E-010	-1.42026E-009	0	1911.18	101.243
600	500	0.6	0.2	1.13156E-009	1.35585E-009	0	2505.2	132.886
700	500	0.6	0.15	-1.57445E-009	6.55461E-010	0	1339.58	91.3363
700	500	0.65	0.15	-2.29108E-010	-4.23682E-010	0	1911.18	108.531
750	600	0.65	0.15	-6.18763E-010	-4.05742E-010	0	1911.18	103.042
750	600	0.65	0.1	-1.07139E-009	-5.42306E-010	0	878.704	48.2649
800	600	0.75	0.1	1.0795E-011	-1.56931E-009	0	878.704	56.0074
800	600	0.75	0.07	-1.09251E-009	-7.97014E-010	0	878.704	36.2319
1000	1000	0.75	0.1	-5.04172E-010	9.11902E-010	0	878.704	52.8373
1000	1200	0.75	0.1	7.8441E-010	-6.25349E-010	0	1143.68	70.4252
1200	1000	0.75	0.1	-1.60801E-009	-1.08215E-009	0	1185.51	62.9488

Tabla 99: Resultados para la función de Rastringin utilizando XHC y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	-9.40731E-010	-1.46587E-009	0	878.704	32.2491
19	3.62727E-010	1.35099E-009	0	878.704	39.8165
43	3.82738E-010	-3.58633E-010	0	444.621	16.0746
83	1.58329E-009	-1.80291E-010	0	878.704	37.8864
101	5.38191E-010	7.33961E-010	0	878.704	34.7705
137	1.31274E-009	1.65997E-009	0	878.704	36.9984
151	-7.81679E-010	2.05841E-010	0	878.704	39.8819
191	-1.85901E-011	5.40905E-010	0	878.704	33.7694
211	7.35815E-010	-4.64029E-010	0	878.704	38.081
233	-2.6216E-010	-1.4334E-009	0	878.704	35
277	1.11851E-009	-8.48775E-010	0	878.704	38.0658
313	5.55828E-010	-2.77564E-010	0	878.704	37.0841
349	0.000000001	-8.7137E-010	0	878.704	37.6339
467	7.32708E-010	-1.1361E-010	0	878.704	38.3995
521	1.04363E-010	-1.52375E-011	0	878.704	36.2298
599	-1.31802E-010	-7.94403E-010	0	878.704	38.4185
617	1.20033E-009	-7.48952E-010	0	878.704	35.4234
661	1.05094E-009	-1.39512E-009	0	878.704	35.2962
683	4.76687E-010	-4.55809E-010	0	878.704	32.7401
701	-1.61594E-010	-7.7728E-010	0	878.704	35.0297
727	1.48047E-009	-6.5045E-010	0	878.704	37.0662
743	-1.53355E-009	6.30802E-010	0	878.704	40.1987
797	1.2257E-009	-3.97263E-010	0	878.704	34.9751
811	-6.09091E-012	-5.10112E-010	0	878.704	38.2229
853	3.16091E-010	-3.27311E-010	0	878.704	37.5065
883	-2.50251E-010	8.20389E-010	0	878.704	33.2028
907	1.51896E-009	1.36716E-009	0	878.704	33.597
991	8.86596E-010	-1.32593E-010	0	878.704	32.6575

Tabla 100: Resultados para la función de Rastrigin utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0
Desviación estándar de los resultados	0

**B.2.21. Rosenbrock & GPAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-0.192201	0.225734	4.98561	4.788490E+008	4.687000E+007
50	50	0.8	0.1	0.702049	0.681783	3.65749	7.398870E+009	2.058550E+008
100	50	0.8	0.1	0.483581	0.20551	0.347007	2.101420E+010	1.493890E+009
150	50	0.8	0.1	1.2472	1.56598	0.0720821	2.541730E+010	1.575340E+009
150	150	0.8	0.1	1.2472	1.56598	0.0720821	2.651450E+010	3.634620E+009
150	150	0.7	0.15	1.2472	1.56598	0.0720821	2.304350E+011	4.155640E+010
150	300	0.7	0.15	1.2472	1.56598	0.0720821	2.411440E+011	4.467250E+010
250	300	0.7	0.15	1.2472	1.56598	0.0720821	1.103430E+012	2.254640E+011
350	300	0.7	0.15	1.2472	1.56598	0.0720821	3.303080E+013	8.150690E+012
400	300	0.7	0.15	0.917176	0.816469	0.0680763	2.611680E+014	2.607680E+013
450	300	0.7	0.15	1.13197	1.29941	0.0499783	9.751350E+013	1.366040E+013
500	300	0.7	0.15	1.13197	1.29941	0.0499783	1.037230E+015	1.273220E+014
500	400	0.7	0.15	1.13197	1.29941	0.0499783	1.217810E+015	1.234710E+014
500	400	0.6	0.15	1.13197	1.29941	0.0499783	5.463290E+018	6.925110E+017
500	400	0.5	0.15	1.13197	1.29941	0.0499783	2.445090E+025	2.934930E+024
500	400	0.5	0.18	1.13197	1.29941	0.0499783	6.817710E+023	1.344920E+023
500	500	0.5	0.18	1.13197	1.29941	0.0499783	2.407310E+026	5.035960E+025
600	500	0.6	0.18	1.1788	1.37656	0.0489287	6.715230E+021	1.530240E+021
600	500	0.6	0.2	1.13197	1.29941	0.0499783	1.128110E+022	1.975610E+021
700	500	0.6	0.15	1.13197	1.29941	0.0499783	7.577620E+023	1.417520E+023
700	500	0.65	0.15	1.13197	1.29941	0.0499783	5.726030E+021	1.124990E+021
750	600	0.65	0.15	1.13197	1.29941	0.0499783	5.580830E+024	7.829330E+023
750	600	0.65	0.1	1.13197	1.29941	0.0499783	1.222660E+024	1.991750E+023
800	600	0.75	0.1	1.13197	1.29941	0.0499783	1.081210E+015	1.342060E+014
800	600	0.75	0.07	1.13197	1.29941	0.0499783	4.388490E+014	6.840520E+013
1000	1000	0.75	0.1	1.13197	1.29941	0.0499783	1.841240E+015	3.202820E+014
1000	1200	0.75	0.1	1.13197	1.29941	0.0499783	1.726650E+015	3.069190E+014
1200	1000	0.75	0.1	1.13197	1.29941	0.0499783	4.219310E+015	5.029980E+014

Tabla 101: Resultados para la función de Rosenbrock utilizando GPAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1.62605	2.64983	0.395281	1.672030E+014	9.065260E+012
19	1.03907	1.0705	0.0099232	1.662030E+014	3.245490E+013
43	0.481043	0.230156	0.269471	7.716640E+014	9.172460E+013
83	0.829973	0.722637	0.143026	1.225840E+014	1.542290E+013
101	1.24818	1.5011	0.384894	5.594190E+014	5.076100E+013
137	0.814493	0.684019	0.0769289	1.604730E+014	2.798980E+013
151	0.950256	0.917646	0.023965	5.164530E+013	1.047030E+013
191	1.35111	1.76734	0.461398	3.874290E+014	6.112380E+013
211	1.37632	1.88074	0.159864	5.540800E+014	2.986760E+013
233	0.763742	0.531479	0.324378	2.178500E+014	3.601050E+013
277	1.10041	1.16753	0.198239	9.913520E+013	2.322590E+013
313	1.16208	1.3726	0.0753889	2.989880E+014	6.322860E+013
349	1.27511	1.58527	0.240746	1.343670E+014	1.697400E+013
467	0.959183	0.905597	0.0225043	1.772590E+014	3.773530E+013
521	1.52583	2.30073	0.351688	7.418170E+013	4.704580E+012
599	1.01613	1.02175	0.0118799	2.159340E+014	3.484750E+013
617	0.67032	0.424561	0.170031	3.050310E+014	3.684520E+013
661	0.877023	0.769752	0.0151572	2.565300E+014	4.433300E+013
683	0.9293	0.873141	0.0141041	1.101160E+014	1.952810E+013
701	1.42236	2.06043	0.317663	3.647260E+014	5.034890E+013
727	1.49626	2.235	0.247709	9.943190E+013	1.544530E+013
743	0.966516	0.917674	0.02828	2.464950E+015	1.797610E+014
797	0.567218	0.314552	0.192462	2.173130E+014	4.128440E+013
811	0.904143	0.823204	0.0124716	3.854860E+014	5.255740E+013
853	0.604821	0.373874	0.162671	1.943570E+014	2.551360E+013
883	1.48416	2.22742	0.295402	4.018120E+014	4.586730E+013
907	0.977976	0.927048	0.0868554	3.528290E+014	6.075390E+013
991	1.1444	1.30383	0.0242311	2.715700E+013	5.763450E+012

Tabla 102: Resultados para la función de Rosenbrock utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	0.0099232
Media de los resultados	0.1684505
Desviación estándar de los resultados	0.1403615

**B.2.22. Rosenbrock & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	1.9033	3.66631	1.00749	1.365080E+009	6.836910E+007
50	50	0.8	0.1	0.967994	0.94597	0.00904692	1.818710E+009	1.855070E+008
100	50	0.8	0.1	1.82543	2.94363	15.7787	2.679870E+010	1.493780E+009
150	50	0.8	0.1	0.704025	0.39167	1.16879	1.000930E+011	6.506170E+009
150	150	0.8	0.1	0.704025	0.39167	1.16879	4.567010E+010	4.660480E+009
150	150	0.7	0.15	2.11615	4.54415	1.68218	1.885570E+011	1.800000E+010
150	300	0.7	0.15	2.11615	4.54415	1.68218	1.560410E+011	2.312810E+010
250	300	0.7	0.15	1.12779	1.22618	0.225478	4.915490E+011	6.452650E+010
350	300	0.7	0.15	2.4168	5.75261	2.78771	1.001910E+012	1.218410E+011
400	300	0.7	0.15	2.75987	7.6261	3.10567	1.083640E+012	1.540030E+011
450	300	0.7	0.15	3.24249	10.3355	8.20468	1.770300E+012	1.789430E+011
500	300	0.7	0.15	3.09714	9.69217	5.39613	1.957880E+012	2.289450E+011
500	400	0.7	0.15	3.09714	9.69217	5.39613	1.197370E+012	1.700950E+011
500	400	0.6	0.15	3.21651	10.3694	4.96792	1.787120E+012	2.556640E+011
500	400	0.5	0.15	0.936206	0.872945	0.00532035	1.744130E+012	2.132670E+011
500	400	0.5	0.18	0.925128	0.816135	0.163427	2.240330E+012	3.555520E+011
500	500	0.5	0.18	0.925128	0.816135	0.163427	2.683520E+012	3.656290E+011
600	500	0.6	0.18	0.652114	0.394709	0.214316	3.792910E+012	5.426390E+011
600	500	0.6	0.2	-1.09189	1.15182	4.53925	4.347490E+012	7.601230E+011
700	500	0.6	0.15	3.22607	10.2681	6.89805	2.792930E+012	3.491750E+011
700	500	0.65	0.15	0.764159	0.638943	0.358164	1.758400E+012	2.891840E+011
750	600	0.65	0.15	1.00414	1.00408	0.00179942	2.397760E+012	3.234240E+011
750	600	0.65	0.1	0.819314	0.639554	0.133271	7.127400E+011	1.022130E+011
800	600	0.75	0.1	0.692643	0.478673	0.0945854	6.623840E+011	8.059990E+010
800	600	0.75	0.07	1.00791	0.977578	0.146818	6.255820E+011	3.018910E+010
1000	1000	0.75	0.1	1.50266	2.2458	0.267527	1.142200E+012	1.062420E+011
1000	1200	0.75	0.1	1.50266	2.2458	0.267527	1.423310E+012	1.548950E+011
1200	1000	0.75	0.1	0.960574	0.84577	0.593402	1.409610E+012	1.742400E+011

Tabla 103: Resultados para la función de Rosenbrock utilizando Recombinación intermedia y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,15). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	2.74605	7.38679	5.41978	2.544960E+012	3.485670E+011
19	1.7232	2.98811	0.557995	2.304270E+012	3.998560E+011
43	0.948529	0.948542	0.24113	1.974440E+012	2.722580E+011
83	0.938659	0.938697	0.335718	2.491210E+012	3.303500E+011
101	1.31581	1.74293	0.113129	3.265560E+012	3.823560E+011
137	0.923033	0.923083	0.51134	2.221020E+012	3.280770E+011
151	1.31726	1.63471	1.10971	2.203400E+012	2.990600E+011
191	0.959037	0.932451	0.0178025	2.089830E+012	3.653750E+011
211	1.08103	1.08104	0.773615	2.879560E+012	3.977570E+011
233	1.07282	1.07286	0.615144	2.373970E+012	4.113710E+011
277	1.06347	1.06347	0.459554	3.766500E+012	3.861460E+011
313	1.00616	1.00616	0.00387802	1.801310E+012	3.223480E+011
349	1.00018	1.07598	0.571971	2.982770E+012	4.027880E+011
467	1.76865	3.11361	0.611849	2.305890E+012	3.723070E+011
521	1.85174	3.41752	0.738531	2.906090E+012	3.752000E+011
599	0.600949	0.320976	0.32056	2.468270E+012	4.660070E+011
617	0.924423	0.924423	0.493831	2.276640E+012	3.102610E+011
661	1.00849	1.00866	0.00711193	2.345750E+012	3.363690E+011
683	1.17395	1.31206	0.46703	2.524570E+012	4.109950E+011
701	0.890513	0.760568	0.117252	3.132710E+012	3.536290E+011
727	0	0	1	3.471410E+012	4.904190E+011
743	0.907605	0.869482	0.217704	2.427520E+012	3.774660E+011
797	0	0	1	2.577850E+012	3.462150E+011
811	2.71761	7.22344	5.57294	2.491320E+012	3.928370E+011
853	1.27816	1.72424	0.897287	2.566990E+012	3.499760E+011
883	1.14888	1.26938	0.277729	2.269400E+012	4.492140E+011
907	0.983929	0.984177	0.0260492	2.921570E+012	3.874140E+011
991	1.06956	1.07666	0.457757	2.709130E+012	3.894960E+011

Tabla 104: Resultados para la función de Rosenbrock utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	0.00387802
Media de los resultados	0.81915706
Desviación estándar de los resultados	1.35789670

**B.2.23. Rosenbrock & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	1.43069	1.10421	89.0496	1.100480E+009	5.502930E+007
50	50	0.8	0.1	1.29065	1.6849	0.121035	1.473120E+009	1.209110E+008
100	50	0.8	0.1	3.76205	14.3558	11.7429	8.319710E+010	3.768110E+009
150	50	0.8	0.1	1.64296	2.76727	0.875047	6.189960E+010	3.290160E+009
150	150	0.8	0.1	1.64296	2.76727	0.875047	1.249790E+011	5.101590E+009
150	150	0.7	0.15	3.41853	11.7235	5.98708	1.955280E+011	2.228980E+010
150	300	0.7	0.15	0.877121	0.788275	0.0509449	1.367360E+011	1.318370E+010
250	300	0.7	0.15	1.02538	1.05299	0.000899006	6.961890E+011	6.996580E+010
350	300	0.7	0.15	1.10576	1.30695	0.720818	2.579130E+012	1.882280E+011
400	300	0.7	0.15	3.41775	11.6545	5.91563	1.783970E+012	1.413630E+011
450	300	0.7	0.15	0.663136	0.448558	0.121237	2.163300E+012	2.604500E+011
500	300	0.7	0.15	-0.867277	0.678172	4.03428	6.333060E+012	1.829030E+011
500	400	0.7	0.15	-0.867277	0.678172	4.03428	2.382010E+012	2.468630E+011
500	400	0.6	0.15	-0.279373	-0.00295123	2.29291	3.874960E+012	3.801370E+011
500	400	0.5	0.15	1.33761	1.77332	0.139233	3.904570E+012	3.811050E+011
500	400	0.5	0.18	0.379792	0.0215886	1.88905	2.707320E+013	7.674430E+011
500	500	0.5	0.18	0.379792	0.0215886	1.88905	1.577380E+013	2.506790E+012
600	500	0.6	0.18	1.27732	1.78204	2.34186	3.791570E+013	3.022540E+012
600	500	0.6	0.2	1.31686	1.79229	0.438707	6.225310E+013	8.168640E+012
700	500	0.6	0.15	2.18796	4.70461	2.09276	1.235590E+013	2.045660E+012
700	500	0.65	0.15	0.885192	0.794011	0.0240947	8.523420E+012	1.448120E+012
750	600	0.65	0.15	0	0	1	7.426490E+013	3.374050E+012
750	600	0.65	0.1	0	0	1	5.300490E+012	6.795350E+011
800	600	0.75	0.1	1.66681	2.76901	0.453174	6.184850E+012	2.418820E+011
800	600	0.75	0.07	1.14493	1.39257	0.688592	1.411990E+013	7.390230E+010
1000	1000	0.75	0.1	1.14445	1.3125	0.0216124	1.426220E+013	8.676950E+011
1000	1200	0.75	0.1	1.14445	1.3125	0.0216124	2.374690E+013	9.461080E+011
1200	1000	0.75	0.1	0.953726	0.891378	0.0353213	3.485110E+013	2.387020E+012

Tabla 105: Resultados para la función de Rosenbrock utilizando SBX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1.16152	1.41739	0.491921	7.794040E+012	6.115210E+011
19	0.661098	0.467989	0.210567	5.593480E+013	6.884640E+011
43	-1.02022	1.03768	4.08227	1.097570E+013	6.617260E+011
83	1.44231	2.02792	0.46951	3.704800E+013	4.914900E+011
101	1.1675	1.40178	0.178091	3.548450E+013	9.618030E+011
137	0.865181	0.78656	0.162746	8.847470E+012	9.542380E+011
151	2.74018	7.45769	3.28703	2.950220E+012	3.516790E+011
191	2.92055	8.52454	3.6911	2.230200E+013	1.332820E+012
211	1.11128	1.17706	0.347347	6.215530E+012	6.649760E+011
233	1.16528	1.41292	0.330297	2.262560E+013	7.131140E+011
277	1.89871	3.62779	0.859163	6.326430E+012	4.774770E+011
313	0.0839358	-0.00605885	0.856345	3.136780E+012	4.710280E+011
349	0.368651	0.157326	0.444494	8.410020E+012	9.044350E+011
467	1.61829	2.6868	0.843693	6.666540E+012	8.444530E+011
521	1.23279	1.45035	0.536082	1.187320E+013	4.945310E+011
599	1.59096	2.50367	0.424682	6.081050E+012	7.281510E+011
617	1.18288	1.41295	0.0523579	1.884640E+013	1.627610E+012
661	1.49257	2.26427	0.375843	6.999320E+012	5.904440E+011
683	0.82661	0.727566	0.226149	6.937870E+014	1.901020E+012
701	1.13215	1.24806	0.131081	8.341400E+012	9.115530E+011
727	2.27389	5.23702	2.06404	4.309630E+012	4.953310E+011
743	0.674908	0.484608	0.190408	6.322760E+012	9.006840E+011
797	1.27485	1.57133	0.366201	4.490440E+014	1.364140E+012
811	0.756054	0.580366	0.0671612	1.028190E+013	7.526780E+011
853	0.154574	0.0805698	1.03597	3.442330E+012	4.506210E+011
883	3.18843	10.2064	4.95167	9.364640E+012	7.015790E+011
907	1.06809	1.17351	0.111515	5.554740E+012	7.237930E+011
991	0.816716	0.658443	0.0409573	3.618480E+012	4.946580E+011

Tabla 106: Resultados para la función de Rosenbrock utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido                    0.040957  
 Media de los resultados                    0.958168  
 Desviación estándar de los resultados    1.351056

**B.2.24. Rosenbrock & SBX Adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0286299	0.175865	4.00764	1.248960E+009	6.244780E+007
50	50	0.8	0.1	0.781664	0.611511	0.0476966	1.311840E+009	1.045320E+008
100	50	0.8	0.1	1.59428	2.54716	0.356126	2.292660E+010	5.648200E+008
150	50	0.8	0.1	1.25478	1.574	0.0649326	2.438090E+010	2.330230E+009
150	150	0.8	0.1	1.2538	1.57298	0.0645086	2.346470E+010	1.900250E+009
150	150	0.7	0.15	0.989662	0.974559	0.00247991	1.482420E+011	9.496850E+009
150	300	0.7	0.15	0.986144	0.971044	0.000398169	3.375200E+011	1.984280E+010
250	300	0.7	0.15	1.21374	1.47867	0.0487069	7.151010E+011	6.601540E+010
350	300	0.7	0.15	1.06306	1.13022	0.00397786	1.157720E+012	4.489950E+010
400	300	0.7	0.15	1.13782	1.29766	0.0199073	1.312220E+012	9.136350E+010
450	300	0.7	0.15	1.15739	1.33785	0.0250589	1.067770E+012	7.051130E+010
500	300	0.7	0.15	1.14598	1.31367	0.0213265	1.941860E+012	1.352830E+011
500	400	0.7	0.15	1.14598	1.31367	0.0213265	9.828790E+011	9.560250E+010
500	400	0.6	0.15	0.90917	0.826412	0.00825323	9.751940E+011	1.041460E+011
500	400	0.5	0.15	1.13218	1.27801	0.0189293	1.679220E+012	5.965680E+010
500	400	0.5	0.18	1.08528	1.18123	0.00842976	1.317180E+012	1.318200E+011
500	500	0.5	0.18	1.08528	1.18123	0.00842976	2.569380E+012	1.481550E+011
600	500	0.6	0.18	1.04249	1.0892	0.00239007	3.376200E+012	3.510980E+011
600	500	0.6	0.2	1.14298	1.30443	0.0208274	3.777940E+012	4.763960E+011
700	500	0.6	0.15	1.147	1.31433	0.0217761	2.361860E+012	2.078440E+011
700	500	0.65	0.15	1.05584	1.11647	0.00339516	1.705490E+012	1.416790E+011
750	600	0.65	0.15	1.05623	1.12896	0.0209666	2.841880E+012	3.207590E+011
750	600	0.65	0.1	1.02379	1.0478	0.000577865	6.221760E+011	3.107640E+010
800	600	0.75	0.1	1.10805	1.23003	0.0121779	7.453660E+011	3.243560E+010
800	600	0.75	0.07	1.06916	1.14437	0.00494397	2.830910E+011	1.422950E+010
1000	1000	0.75	0.1	1.02519	1.05132	0.000643655	1.255140E+012	8.872930E+010
1000	1200	0.75	0.1	1.02519	1.05132	0.000643655	1.311660E+012	8.636010E+010
1200	1000	0.75	0.1	1.02256	1.04426	0.000694125	1.059900E+012	1.138850E+011

Tabla 107: Resultados para la función de Rosenbrock utilizando SBX Adaptativo y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	0.957141	0.914599	0.00206818	2.568670E+012	9.007540E+010
19	1.02991	1.06286	0.00135408	9.820130E+011	5.487890E+010
43	1.0601	1.12422	0.00362855	7.429410E+011	4.019360E+010
83	0.864649	0.745338	0.0188396	6.121980E+011	4.862470E+010
101	1.01135	1.02094	0.000482066	8.324930E+011	6.626240E+010
137	0.951845	0.90468	0.00249555	8.231960E+011	6.195160E+010
151	0.958044	0.917705	0.00176234	9.333470E+011	7.396640E+010
191	1.05707	1.10645	0.0152193	1.164010E+012	5.296220E+010
211	1.05211	1.11017	0.00375381	7.045930E+011	5.389140E+010
233	0.981343	0.961968	0.000461691	8.873700E+011	5.700930E+010
277	1.04767	1.09855	0.00235899	1.993750E+012	1.323290E+011
313	0.934689	0.874348	0.00431534	1.000690E+012	6.103070E+010
349	1.17994	1.39322	0.0324709	1.816720E+012	1.154250E+011
467	1.02406	1.04806	0.000617991	6.730820E+011	6.746940E+010
521	0.952758	0.906175	0.00247936	7.817550E+011	6.049090E+010
599	1.20752	1.4615	0.0442174	5.634860E+011	4.174110E+010
617	1.06833	1.14358	0.00517365	1.494480E+012	7.075950E+010
661	1.08934	1.18801	0.00816157	8.879130E+011	4.804160E+010
683	0.870414	0.759793	0.0172651	6.996870E+011	7.411240E+010
701	1.02024	1.04144	0.000439409	1.321560E+012	6.516500E+010
727	1.00681	1.01396	5.52467E-005	6.896430E+011	5.895670E+010
743	0.917709	0.844235	0.00719012	7.800350E+011	5.317750E+010
797	1.09194	1.19489	0.00910677	9.271250E+011	6.743030E+010
811	1.05527	1.12174	0.00968762	9.592060E+011	7.586750E+010
853	0.934567	0.879797	0.00835439	9.386200E+011	6.582840E+010
883	0.970592	0.937592	0.00285114	1.206240E+012	7.954640E+010
907	1.05374	1.11374	0.00402788	7.167890E+011	5.900410E+010
991	1.13021	1.28179	0.018913	8.483890E+011	5.971720E+010

Tabla 108: Resultados para la función de Rosenbrock utilizando SBX Adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido	5.52467E-005
Media de los resultados	0.00813397
Desviación estándar de los resultados	0.01034485

**B.2.25. Rosenbrock & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	0.0448399	0.00178524	0.912336	1.252070E+009	6.260330E+007
50	50	0.8	0.1	0.999999	0.999999	3.56787E-013	1.339560E+009	1.071650E+008
100	50	0.8	0.1	1	1	3.10888E-014	1.339560E+009	1.071630E+008
150	50	0.8	0.1	1	1	5.02334E-016	2.380680E+010	6.425730E+008
150	150	0.8	0.1	1	1	1.10934E-029	2.108310E+010	5.899810E+008
150	150	0.7	0.15	1	1	4.93038E-032	2.108310E+010	8.247230E+008
150	300	0.7	0.15	1	1	0	2.108310E+010	8.382150E+008
250	300	0.7	0.15	1	1	0	4.246050E+010	1.948250E+009
350	300	0.7	0.15	1	1	0	4.155360E+010	1.656820E+009
400	300	0.7	0.15	1	1	0	1.061460E+011	1.972640E+009
450	300	0.7	0.15	1	1	3.08149E-031	1.061460E+011	2.390850E+009
500	300	0.7	0.15	1	1	4.43734E-031	1.642230E+011	2.652840E+009
500	400	0.7	0.15	1	1	0	4.451470E+010	1.767210E+009
500	400	0.6	0.15	1	1	0	1.061460E+011	1.678480E+009
500	400	0.5	0.15	1	1	0	3.901480E+010	9.108210E+008
500	400	0.5	0.18	1	1	0	1.061460E+011	1.603110E+009
500	500	0.5	0.18	1	1	0	4.120820E+010	1.520780E+009
600	500	0.6	0.18	1	1	0	1.147730E+011	2.369010E+009
600	500	0.6	0.2	1	1	0	1.061500E+011	2.891670E+009
700	500	0.6	0.15	1	1	0	1.725040E+011	1.735590E+009
700	500	0.65	0.15	1	1	0	1.137340E+011	2.696890E+009
750	600	0.65	0.15	1	1	0	1.345510E+011	2.447090E+009
750	600	0.65	0.1	1	1	0	1.061460E+011	9.510160E+008
800	600	0.75	0.1	1	1	0	1.061460E+011	1.622400E+009
800	600	0.75	0.07	1	1	0	2.108310E+010	4.882640E+008
1000	1000	0.75	0.1	1	1	0	1.061460E+011	1.675780E+009
1000	1200	0.75	0.1	1	1	0	1.061460E+011	1.649050E+009
1200	1000	0.75	0.1	1	1	0	4.501280E+010	1.506470E+009

Tabla 109: Resultados para la función de Rosenbrock utilizando XHC y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (750), el número de generaciones (600), la tasa de supervivencia (0,65) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	1	1	0	1.061460E+011	8.180800E+008
19	1	1	0	2.108360E+010	5.328990E+008
43	1	1	0	2.108310E+010	6.888350E+008
83	1	1	0	3.859630E+010	8.432900E+008
101	1	1	0	3.938040E+010	8.161620E+008
137	1	1	0	1.061460E+011	9.623800E+008
151	1	1	0	2.489240E+010	6.518010E+008
191	1	1	0	3.723750E+010	7.695590E+008
211	1	1	0	4.060770E+010	7.446800E+008
233	1	1	0	3.565720E+010	7.901120E+008
277	1	1	0	2.296440E+010	6.653620E+008
313	1	1	0	2.108310E+010	6.416520E+008
349	1	1	0	2.108310E+010	8.866000E+008
467	1	1	0	4.547970E+010	8.932620E+008
521	1	1	0	2.108310E+010	5.883830E+008
599	1	1	0	2.370320E+010	6.200690E+008
617	1	1	0	1.061460E+011	1.156010E+009
661	1	1	0	1.154980E+011	1.635740E+009
683	1	1	0	2.108310E+010	5.913250E+008
701	1	1	0	2.108320E+010	5.403340E+008
727	1	1	0	3.335440E+010	5.676960E+008
743	1	1	0	1.718040E+011	9.595110E+008
797	1	1	0	3.089760E+010	8.782770E+008
811	1	1	0	2.108310E+010	5.634800E+008
853	1	1	0	3.190610E+010	7.639420E+008
883	1	1	0	4.427600E+010	7.245330E+008
907	1	1	0	4.145710E+010	7.400510E+008
991	1	1	0	3.544410E+010	8.181870E+008

Tabla 110: Resultados para la función de Rosenbrock utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido	0
Media de los resultados	0
Desviación estándar de los resultados	0

**B.2.26. Schwefel & GPAX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-142.312	394.264	420.189	944.734	686.498
50	50	0.8	0.1	419.619	421.41	0.254314	925.297	111.079
100	50	0.8	0.1	420.45	420.57	0.0540068	898.821	103.219
150	50	0.8	0.1	421.59	419.707	0.249528	972.331	174.926
150	150	0.8	0.1	421.576	419.702	0.248811	969.39	178.546
150	150	0.7	0.15	419.267	420.896	0.365981	1419.93	201.467
150	300	0.7	0.15	419.267	420.896	0.365981	1107.47	309.379
250	300	0.7	0.15	421.006	421.042	0.000872427	1399.73	396.15
350	300	0.7	0.15	420.669	420.339	0.0614589	1218.23	388.471
400	300	0.7	0.15	420.772	420.956	0.00494478	1402.32	397.391
450	300	0.7	0.15	420.725	421.076	0.00895195	1418.93	463.235
500	300	0.7	0.15	420.787	420.991	0.00425056	1308.18	403.879
500	400	0.7	0.15	420.787	420.976	0.00419462	1209.02	420.676
500	400	0.6	0.15	421.451	420.323	0.0819407	1420.66	414.968
500	400	0.5	0.15	421.141	420.607	0.0202481	1372.55	618.283
500	400	0.5	0.18	421.241	420.863	0.0107657	1430.4	586.876
500	500	0.5	0.18	421.166	420.808	0.00816723	1400.92	631.083
600	500	0.6	0.18	420.824	421.039	0.00327133	1430.71	541.656
600	500	0.6	0.2	420.481	421.417	0.0553634	1431.88	559.253
700	500	0.6	0.15	420.719	420.99	0.00796999	1411.76	444.072
700	500	0.65	0.15	421.576	421.15	0.0507185	1402.26	495.611
750	600	0.65	0.15	420.837	420.991	0.00228421	1433.19	495.099
750	600	0.65	0.1	420.718	421.035	0.00851567	1386.55	393.019
800	600	0.75	0.1	420.982	420.773	0.00489193	1392.3	363.557
800	600	0.75	0.07	420.867	420.992	0.00140869	1162.43	288.529
1000	1000	0.75	0.1	420.707	420.199	0.0833969	1368.41	369.451
1000	1200	0.75	0.1	420.707	420.199	0.0833969	1274.95	335.049
1200	1000	0.75	0.1	420.212	420.587	0.0906246	1328.59	333.391

Tabla 111: Resultados para la función de Schwefel utilizando GPAX y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.64	420.599	0.0309383	1342.44	332.623
19	420.884	421.019	0.00124123	1427.34	304.049
43	420.603	421.428	0.0434331	1415.18	302.815
83	420.502	421.305	0.0417549	1256.61	308.983
101	420.805	421.061	0.00447102	1378.9	331.85
137	420.994	420.506	0.027084	1430.1	319.05
151	421.506	420.412	0.0756192	1409.99	300.707
191	421.122	420.631	0.0173664	1427.1	365.374
211	421.441	420.041	0.136759	1243.3	317.897
233	420.89	420.991	0.000866034	1419.3	350.108
277	421.025	420.904	0.000954806	1359.65	315.61
313	421.022	420.762	0.00576254	1388.36	312.305
349	420.849	420.972	0.0018314	1415.6	317.972
467	421.483	419.771	0.214274	1412.86	383.662
521	421.014	420.847	0.00215796	1415.96	341.46
599	421.388	420.7	0.0313098	1397.37	362.462
617	420.893	421.01	0.00097209	1430.64	355.113
661	420.666	420.464	0.0437571	1375.26	317.573
683	420.654	420.464	0.0447296	1328.89	294.886
701	420.444	421.19	0.0409281	1429.23	338.934
727	420.97	420.833	0.00234158	1412.7	299.399
743	420.868	420.652	0.0139903	1434.83	342.347
797	421.031	420.834	0.00280418	1270.01	293.948
811	420.486	420.395	0.0709393	1193.88	300.807
853	420.647	421.047	0.0138623	1411	354.664
883	420.911	420.962	0.000458476	1395.15	359.394
907	420.681	420.983	0.0105175	1436.08	332.278
991	420.862	418.977	0.501707	1416.21	316.012

Tabla 112: Resultados para la función de Schwefel utilizando GPAX y una semilla de aleatorios variable.

Mejor resultado obtenido	0.00045848
Media de los resultados	0.04938683
Desviación estándar de los resultados	0.10028732

**B.2.27. Schwefel & Recombinación intermedia**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	442.189	426.215	59.4971	736.579	92.7971
50	50	0.8	0.1	421.549	417.38	1.66423	840.785	74.3293
100	50	0.8	0.1	421.205	421.209	0.0143778	841.534	71.1639
150	50	0.8	0.1	421.294	421.123	0.0164215	836.953	84.1054
150	150	0.8	0.1	421.055	420.998	0.00106365	1126.7	70.5324
150	150	0.7	0.15	420.925	420.953	0.000294153	1131.25	128.303
150	300	0.7	0.15	420.925	420.953	0.000294153	1131.31	132.41
250	300	0.7	0.15	421.189	420.404	0.0463433	1131.32	148.771
350	300	0.7	0.15	420.965	420.939	0.000136351	1148.46	163.894
400	300	0.7	0.15	421.096	420.768	0.00712323	1137.93	162.26
450	300	0.7	0.15	420.971	420.944	0.000100729	1136.95	164.469
500	300	0.7	0.15	420.896	420.643	0.0141126	1136.65	164.964
500	400	0.7	0.15	420.896	420.643	0.0141126	1138.63	171.859
500	400	0.6	0.15	420.811	420.994	0.00323531	1139.36	185.65
500	400	0.5	0.15	421.298	420.431	0.0501838	1192.94	173.671
500	400	0.5	0.18	421.061	420.185	0.0785581	1143.8	216.121
500	500	0.5	0.18	421.061	420.185	0.0785581	1146.55	225.892
600	500	0.6	0.18	420.988	420.93	0.000259171	1160.88	217.544
600	500	0.6	0.2	420.854	420.863	0.00309665	1146.06	258.295
700	500	0.6	0.15	420.979	420.939	0.000151807	1151.18	204.284
700	500	0.65	0.15	420.943	420.976	0.000114476	1139.67	172.86
750	600	0.65	0.15	421.07	420.885	0.00220921	1140.01	180.964
750	600	0.65	0.1	420.73	421.187	0.0132558	1139.04	137.487
800	600	0.75	0.1	420.886	420.85	0.00266043	1137.39	114.872
800	600	0.75	0.07	420.907	420.981	0.00052773	1133.47	80.3236
1000	1000	0.75	0.1	420.976	420.961	4.12867E-005	1134.78	119.952
1000	1200	0.75	0.1	420.976	420.961	4.12867E-005	1136.52	127.84
1200	1000	0.75	0.1	420.969	420.933	0.000187526	1136.7	121.685

Tabla 113: Resultados para la función de Schwefel utilizando Recombinación intermedia y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (1000), el número de generaciones (1000), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.986	420.953	9.62682E-005	1171.2	116.851
19	420.955	420.952	8.29205E-005	1141.09	123.498
43	420.961	420.979	4.59612E-005	1164.17	111.143
83	420.829	420.905	0.00298981	1136.05	128.231
101	420.967	420.96	3.52201E-005	1137.4	122.563
137	421.016	420.924	0.000566205	1136.44	126.667
151	420.969	420.872	0.00121714	1139.13	120.589
191	420.998	420.91	0.000564741	1138.7	118.581
211	420.927	420.9	0.000837503	1153.07	118.97
233	421.103	420.84	0.00441102	1144.5	127.303
277	421.044	420.915	0.00110443	1136.11	118.058
313	420.984	420.948	0.000108066	1135.7	132.527
349	420.929	420.983	0.000246488	1137.88	121.593
467	420.901	420.914	0.000975215	1137.43	118.722
521	420.92	420.987	0.000372472	1136.13	123.511
599	421.023	420.936	0.000530242	1135.38	121.483
617	420.991	420.872	0.00125584	1136.64	100.612
661	420.97	420.966	2.68148E-005	1138.51	124.981
683	421.067	420.849	0.0030616	1135.75	127.794
701	421.016	420.841	0.00236922	1140.42	124.003
727	420.895	421.005	0.000881574	1144.23	125.583
743	420.96	420.975	4.04285E-005	1135.39	120.447
797	420.934	420.983	0.000204468	1135.64	111.468
811	420.915	421.041	0.00104694	1138.52	116.634
853	420.968	420.964	2.87285E-005	1138.46	124.59
883	420.936	420.984	0.000187337	1142.57	126.202
907	420.977	420.914	0.000412231	1136.09	119.625
991	420.751	421.26	0.0167145	1138.56	126.195

Tabla 114: Resultados para la función de Schwefel utilizando Recombinación intermedia y una semilla de aleatorios variable.

Mejor resultado obtenido	2.68148E-005
Media de los resultados	0.00144333
Desviación estándar de los resultados	0.00318347

**B.2.28. Schwefel & SBX**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	415.918	-319.799	159.004	1008.22	235.043
50	50	0.8	0.1	421.228	420.81	0.011658	839.602	68.2536
100	50	0.8	0.1	422.736	419.541	0.651366	849.523	76.2165
150	50	0.8	0.1	420.908	421.007	0.000675935	911.794	97.8967
150	150	0.8	0.1	420.926	420.965	0.000253823	1132.15	72.2
150	150	0.7	0.15	420.955	420.975	5.34048E-005	1130.06	159.937
150	300	0.7	0.15	420.963	420.974	3.23314E-005	911.997	122.149
250	300	0.7	0.15	420.984	420.961	6.13941E-005	1141.74	149.957
350	300	0.7	0.15	420.962	420.98	4.76151E-005	1182.82	175.043
400	300	0.7	0.15	420.955	420.97	4.86952E-005	1131.91	163.378
450	300	0.7	0.15	420.968	420.965	2.72086E-005	1147	174.777
500	300	0.7	0.15	420.981	420.966	4.50319E-005	1140.2	175.69
500	400	0.7	0.15	420.979	420.964	4.26238E-005	1142.74	177.113
500	400	0.6	0.15	420.964	420.968	2.89096E-005	1134.3	170.721
500	400	0.5	0.15	420.985	420.966	6.07924E-005	1205.26	173.363
500	400	0.5	0.18	420.974	420.965	3.02776E-005	1188.92	234.728
500	500	0.5	0.18	420.974	420.965	3.02776E-005	1289.5	226.908
600	500	0.6	0.18	420.965	420.969	2.71923E-005	1210.55	219.392
600	500	0.6	0.2	420.968	420.988	7.30043E-005	1435.06	270.593
700	500	0.6	0.15	420.965	420.973	2.91567E-005	1268.43	215.794
700	500	0.65	0.15	420.963	420.974	3.25774E-005	1306.16	196.757
750	600	0.65	0.15	420.968	420.972	2.70266E-005	1190.52	187.426
750	600	0.65	0.1	420.972	420.965	2.79571E-005	1192.3	113.587
800	600	0.75	0.1	420.966	420.97	2.68062E-005	1328.97	120.681
800	600	0.75	0.07	420.969	420.972	2.65126E-005	1133.37	86.7013
1000	1000	0.75	0.1	420.966	420.971	0.000027694	1190.53	139.914
1000	1200	0.75	0.1	420.966	420.971	0.000027694	1177.08	118.376
1200	1000	0.75	0.1	420.969	420.971	0.000026193	1328.74	129.504

Tabla 115: Resultados para la función de Schwefel utilizando SBX y una semilla de aleatorios fija.



**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.972	420.966	2.78008E-005	1317.33	88.0463
19	420.97	420.961	3.33129E-005	1129.06	84.4498
43	420.97	420.97	2.56874E-005	1126.12	82.8975
83	420.979	420.952	7.37452E-005	1137.28	84.6353
101	420.967	420.968	2.58302E-005	906.459	80.4795
137	420.968	420.968	0.00002571	1142.94	88.3166
151	420.971	420.966	2.74474E-005	1193.54	81.4124
191	420.966	420.974	3.00028E-005	914.677	84.361
211	420.965	420.97	2.73704E-005	1135.13	82.377
233	420.968	420.969	2.55322E-005	1044.64	80.0286
277	420.969	420.969	2.54609E-005	1140.03	91.7475
313	420.962	420.965	3.26801E-005	1133.48	88.7375
349	420.966	420.972	2.77482E-005	1130.11	83.8516
467	420.962	420.972	3.34234E-005	1121.4	87.9741
521	420.971	420.964	0.000028384	1420.14	89.5542
599	420.966	420.982	4.87374E-005	1134.62	89.5082
617	420.965	420.969	2.69272E-005	1129.5	82.5748
661	420.972	420.966	2.78387E-005	1130.29	78.1866
683	420.967	420.971	0.000026496	891.104	76.085
701	420.964	420.971	2.94453E-005	1110.27	88.3048
727	420.97	420.967	2.57726E-005	1129.72	77.9039
743	420.97	420.97	2.57322E-005	1207.27	87.1542
797	420.966	420.968	2.63024E-005	1154.55	97.6102
811	420.965	420.972	0.000028496	1133.56	73.6741
853	420.969	420.973	2.75455E-005	1129	81.8103
883	420.969	420.968	2.54684E-005	1183.46	87.7216
907	420.969	420.962	3.07561E-005	1026.12	85.8525
991	420.965	420.969	2.73199E-005	1284.32	87.4805

Tabla 116: Resultados para la función de Schwefel utilizando SBX y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54609E-005
Media de los resultados	3.024905E-005
Desviación estándar de los resultados	0.00000967

**B.2.29. Schwefel & SBX Adaptativo**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	2.78963	5.02016	831.254	904.313	793.871
50	50	0.8	0.1	215.009	214.584	464.905	1438.57	544.302
100	50	0.8	0.1	384.024	384.38	313.343	1011.27	379.006
150	50	0.8	0.1	396.235	396.456	146.899	957.08	227.398
150	150	0.8	0.1	397.132	397.357	136.73	1152.03	209.319
150	150	0.7	0.15	397.873	395.984	140.44	1154.74	291.275
150	300	0.7	0.15	404.579	402.493	75.2464	950.13	199.718
250	300	0.7	0.15	421.038	420.979	0.000639213	1138.5	153.695
350	300	0.7	0.15	407.639	407.44	44.8642	1147.8	228.597
400	300	0.7	0.15	410.31	405.458	44.0023	1149	213.142
450	300	0.7	0.15	421.182	420.768	0.010848	1139.42	165.311
500	300	0.7	0.15	410.724	410.987	25.5813	1147.29	210.273
500	400	0.7	0.15	415.575	415.855	6.94761	1139.54	175.537
500	400	0.6	0.15	420.815	421.123	0.00598561	1130	142.815
500	400	0.5	0.15	403.742	433.932	57.7715	1191.26	205.992
500	400	0.5	0.18	420.855	421.082	0.00328498	1132.74	182.048
500	500	0.5	0.18	420.855	421.082	0.00328471	1130.62	194.283
600	500	0.6	0.18	420.898	421.039	0.00127901	1191.43	182.368
600	500	0.6	0.2	421.178	420.759	0.011088	1132	204.477
700	500	0.6	0.15	420.995	420.943	0.000195381	1134.38	157.29
700	500	0.65	0.15	420.84	421.097	0.00418223	1190.39	155.337
750	600	0.65	0.15	420.977	420.962	4.02355E-005	1190.2	159.596
750	600	0.65	0.1	420.972	420.965	2.86205E-005	1135.36	109.272
800	600	0.75	0.1	420.825	421.112	0.00520768	1136.76	108.388
800	600	0.75	0.07	420.511	421.422	0.0524652	1137.51	85.9163
1000	1000	0.75	0.1	420.661	421.274	0.0237223	1130.76	109.929
1000	1200	0.75	0.1	420.661	421.274	0.0237223	1131.36	109.324
1200	1000	0.75	0.1	421.185	420.751	0.0118864	1129.49	109.716

Tabla 117: Resultados para la función de Schwefel utilizando SBX Adaptativo y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,10). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	421.158	420.78	0.00907315	1128.15	101.115
19	421.195	420.742	0.0129832	1130.93	116.495
43	420.969	420.969	2.54551E-005	1128.33	107.203
83	420.901	421.037	0.001197	1128.45	109.153
101	420.967	420.971	2.62864E-005	1127.45	112.369
137	421.111	420.826	0.00514747	1128.28	121.982
151	420.973	420.965	2.96607E-005	1128.28	123.822
191	420.712	421.223	0.0164806	1130.77	114.411
211	421.325	420.606	0.0326027	1128.84	113.451
233	421.291	420.652	0.0257689	1190.52	110.506
277	420.669	421.266	0.0225051	1133.16	114
313	420.866	421.074	0.00274927	1130.9	104.053
349	420.977	420.961	4.16627E-005	1128.25	111.281
467	420.826	421.113	0.00520759	1129.38	105.193
521	421.133	420.81	0.00660087	1131.17	103.477
599	420.93	421.007	0.000392526	1128.36	113.363
617	421.093	420.844	0.00392503	1131.97	104.763
661	420.721	421.216	0.015481	1190.36	96.3728
683	420.368	421.548	0.0879713	1129.18	103.029
701	415.793	425.564	6.03586	1129	108.066
727	420.804	421.134	0.00690338	1130.6	100.511
743	421.109	420.84	0.00458744	1129.5	100.438
797	420.735	421.202	0.013769	1133.09	108.844
811	421.105	420.835	0.00461286	1134.23	100.959
853	421.078	420.859	0.00306075	1129.87	112.606
883	420.154	421.725	0.155879	1128.35	103.108
907	420.866	421.071	0.00266519	1129.58	107.194
991	420.867	421.07	0.00261191	1133.18	105.501

Tabla 118: Resultados para la función de Schwefel utilizando SBX Adaptativo y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54551E-005
Media de los resultados	0.23136280
Desviación estándar de los resultados	1.13803369

**B.2.30. Schwefel & XHC**

**Primer conjunto de pruebas.** Se mantiene una semilla de aleatorios constante (100), variando en cada prueba el tamaño de la población, número de generaciones, tasa de supervivencia y tasa de mutación.

$T_{pop}$	$G$	$T_s$	$T_m$	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
20	20	0.8	0.1	-307.723	420.777	121.868	193.571	119.586
50	50	0.8	0.1	-302.525	420.969	118.438	216.996	123.953
100	50	0.8	0.1	420.969	420.969	2.54551E-005	99.7518	7.97853
150	50	0.8	0.1	420.969	420.969	2.54551E-005	99.7286	8.64316
150	150	0.8	0.1	420.969	420.969	2.54551E-005	99.7286	8.64316
150	150	0.7	0.15	420.969	420.969	2.54551E-005	379.101	27.5277
150	300	0.7	0.15	420.969	420.969	2.54551E-005	379.101	21.1434
250	300	0.7	0.15	420.969	420.969	2.54551E-005	379.101	38.3085
350	300	0.7	0.15	420.969	420.969	2.54551E-005	379.101	30.262
400	300	0.7	0.15	420.969	420.969	2.54551E-005	379.101	35.1732
450	300	0.7	0.15	420.969	420.969	2.54551E-005	379.101	35.6283
500	300	0.7	0.15	420.969	420.969	2.54551E-005	780.141	53.3859
500	400	0.7	0.15	420.969	420.969	2.54551E-005	379.101	35.2273
500	400	0.6	0.15	420.969	420.969	2.54551E-005	379.101	30.4007
500	400	0.5	0.15	420.969	420.969	2.54551E-005	379.101	31.0459
500	400	0.5	0.18	420.969	420.969	2.54551E-005	379.101	34.7095
500	500	0.5	0.18	420.969	420.969	2.54551E-005	379.101	41.3
600	500	0.6	0.18	420.969	420.969	2.54551E-005	780.141	55.8237
600	500	0.6	0.2	420.969	420.969	2.54551E-005	1010.18	63.4839
700	500	0.6	0.15	420.969	420.969	2.54551E-005	780.141	42.6268
700	500	0.65	0.15	420.969	420.969	2.54551E-005	780.141	54.8114
750	600	0.65	0.15	420.969	420.969	2.54551E-005	780.141	43.7461
750	600	0.65	0.1	420.969	420.969	2.54551E-005	780.141	20.4822
800	600	0.75	0.1	420.969	420.969	2.54551E-005	379.101	24.062
800	600	0.75	0.07	420.969	420.969	2.54551E-005	379.101	18.7751
1000	1000	0.75	0.1	420.969	420.969	2.54551E-005	379.101	29.2304
1000	1200	0.75	0.1	420.969	420.969	2.54551E-005	379.101	33.7335
1200	1000	0.75	0.1	420.969	420.969	2.54551E-005	780.141	36.4743

Tabla 119: Resultados para la función de Schwefel utilizando XHC y una semilla de aleatorios fija.

**Segundo conjunto de pruebas.** Se mantienen fijos el tamaño de la población (800), el número de generaciones (600), la tasa de supervivencia (0,75) y la tasa de mutación (0,07). Para cada prueba se utiliza como semilla aleatoria un número primo distinto.

Semilla	$x_1$	$x_2$	$F(x_1, x_2)$	Peor	Promedio
7	420.969	420.969	2.54551E-005	379.101	16.2323
19	420.969	420.969	2.54551E-005	379.101	17.0051
43	420.969	420.969	2.54551E-005	379.101	16.4808
83	420.969	420.969	2.54551E-005	379.101	15.6833
101	420.969	420.969	2.54551E-005	379.101	16.5064
137	420.969	420.969	2.54551E-005	379.101	15.259
151	420.969	420.969	2.54551E-005	379.101	16.4817
191	420.969	420.969	2.54551E-005	379.101	16.0565
211	420.969	420.969	2.54551E-005	379.101	16.8804
233	420.969	420.969	2.54551E-005	379.101	15.9822
277	420.969	420.969	2.54551E-005	379.101	15.8575
313	420.969	420.969	2.54551E-005	379.101	16.6558
349	420.969	420.969	2.54551E-005	379.101	17.6284
467	420.969	420.969	2.54551E-005	379.101	16.281
521	420.969	420.969	2.54551E-005	780.141	15.8602
599	420.969	420.969	2.54551E-005	379.101	16.5807
617	420.969	420.969	2.54551E-005	379.101	18.0766
661	420.969	420.969	2.54551E-005	379.101	16.6558
683	420.969	420.969	2.54551E-005	379.101	14.0619
701	420.969	420.969	2.54551E-005	379.101	16.007
727	420.969	420.969	2.54551E-005	379.101	15.259
743	420.969	420.969	2.54551E-005	379.101	16.9547
797	420.969	420.969	2.54551E-005	379.101	13.2644
811	420.969	420.969	2.54551E-005	379.101	15.3836
853	420.969	420.969	2.54551E-005	780.141	15.8354
883	420.969	420.969	2.54551E-005	379.101	14.1123
907	420.969	420.969	2.54551E-005	379.101	16.0317
991	420.969	420.969	2.54551E-005	379.101	18.501

Tabla 120: Resultados para la función de Schwefel utilizando XHC y una semilla de aleatorios variable.

Mejor resultado obtenido	2.54551E-005
Media de los resultados	2.54551E-005
Desviación estándar de los resultados	0

## C. Código fuente

### C.1. Algoritmo genético

A continuación se muestra el código fuente de las partes que componen el algoritmo genético simple.

#### C.1.1. Generación de la población inicial

##### Codificación binaria .

```
/* Generación de la población inicial.
Se genera un número aleatorio entre 0 y 100. Dependiendo si el valor
es menor o mayor a 50 se le da valor al bit
*/
for (i = 0 ; i < Tpop ; i++){
    for (j = 0 ; j < BITS ; j++){
        if ((rand() % 100) < 50)
            auxiliar[j] = 0;
        else
            auxiliar[j] = 1;
    }
    x.push_back(auxiliar);
}
for (i = 0 ; i < Tpop ; i++){
    for (j = 0 ; j < BITS ; j++){
        if ((rand() % 100) < 50)
            auxiliar[j] = 1;
        else
            auxiliar[j] = 0;
    }
    y.push_back(auxiliar);
}
```

##### Codificación real .

```
/*LS indica el limite superior del rango de las variables. LI define el
limite inferior del rango*/
for (i = 0 ; i < Tpop ; i++){
    x1[i] = LS - double_rand() * (LS - LI);
    x2[i] = LS - double_rand() * (LS - LI);
    r[i] = FO(x1[i], x2[i]);
}
```

### C.1.2. Selección de los supervivientes

De acuerdo con el valor indicado en la Tasa de supervivencia,  $T_s$ , se selecciona el número indicado de supervivientes que pasaran a la siguiente generación. Se utilizo el método de selección conocido como SUS (Stochastic universal sampling [31]) . Se utiliza el mismo método para la selección de padres para la cruce.

```
//Selección de supervivientes (SUS)
apttotal = 0.0;
for ( i = 0 ; i < Tpop ; i++){
    double dif = peorindividuo - fobj[i];
    apt.push_back(dif);
    apttotal = apttotal + dif;
}
for ( i = 0 ; i < Tpop ; i++){
    double rel = apt[i] / apttotal;
    aptrel.push_back(rel);
}
ptr = double_rand() / nsel;
suma = 0.0;
i = 0;
j = 0;
while ( i < nsel){
    suma = suma + aptrel[j];
    while (suma > ptr){
        newx.push_back(x[j]);
        newy.push_back(y[j]);
        i++;
        ptr = ptr + 1.0/nsel;
    }
    j++;
}
}
```

### C.1.3. Mutación

**Codificación binaria.** Se utilizó la técnica de mutación conocida como *bitflip*. En esta técnica la tasa de mutación  $T_m$  establece la probabilidad de invertir el valor un determinado gen en cada uno de los miembros de la población.

```
for ( i = 0 ; i < Tpop ; i++){
    for ( j = 0 ; j < BITS ; j++){
        mutador = double_rand();
        if ( mutador < Tm)
            newx[i].flip(j);
        mutador = double_rand();
        if ( mutador < Tm)
            newy[i].flip(j);
    }
}
}
```

**Codificación real.** Se utilizó la técnica de mutación conocida como *mutación paramétrica*. Esta técnica propuesta por Deb [26], establece el número de individuos a mutar con respecto al valor de  $T_m$ , los cuales sufren una variación mediante un valor  $\sigma$  que se obtiene mediante una función similar a la utilizada en la técnica de cruce SBX.

```

nm = 100 + t;
nmut = Tm * Tpop;
delta = LS - LI;
for (i = 0 ; i < nmut ; i++){
    u = double_rand();

    if (u <= 0.5)
        sigma = pow((2*u) , 1/(nm+1));
    else
        sigma = 1 - pow((2*(1-u)) , 1/(nm+1));

    ms = rand() % Tpop;

    seleccionx1[ms] = seleccionx1[ms] + delta * sigma;
    seleccionx2[ms] = seleccionx2[ms] + delta * sigma;
}
    
```

#### C.1.4. Elitismo

```

mejor = 0; //Busqueda de la mejor solución en la población actual
for (i = 1 ; i < Tpop ; i++){
    if (r[i] < r[mejor])
        mejor = i;
}

for (i = 0 ; i < Tpop ; i++){ //Evaluación de la nueva población
    r[i] = FO(seleccionx1[i], seleccionx2[i]);
}

peor = 0; //Busqueda de la peor solución en la nueva población
for (i = 1 ; i < Tpop ; i++){
    if (r[i] > r[peor])
        peor = i;
}
seleccionx1[peor] = x1[mejor]; //Sustitución de los datos
seleccionx2[peor] = x2[mejor];
r[peor] = FO(x1[mejor], x2[mejor]);
x1 = seleccionx1;
x2 = seleccionx2;
    
```



## C.2. Funciones de prueba

En esta sección se presenta el código fuente de las funciones de prueba utilizadas. En todos los casos se utilizaron dos variables.

### C.2.1. Ackley

```
#define PI 3.1416

double FO(double x1, double x2){
    double r;
    double a = (pow(x1,2) + pow(x2,2)) / 2;
    double b = (cos(2 * PI * x1) + cos(2 * PI * x2))/2;
    r = 20 + exp(1) - 20 * exp(-0.2 * sqrt(a)) - exp(b);
    return r;
}
```

### C.2.2. Esfera

```
double FO(double x1, double x2){
    double r;
    r = pow(x1,2) + pow(x2,2);
    return r;
}
```

### C.2.3. Griewank

```
double FO(double x1, double x2){
    double r;
    double s = (pow(x1, 2) + pow(x2, 2))/4000;
    double p = cos(x1) * cos(x2/sqrt(2));
    r = s - p + 1;
    return r;
}
```

### C.2.4. Rastringin

```
double FO(double x1, double x2){
    double r;
    r = pow(x1,2) - 10 * cos(2 * PI * x1) + 10;
    r += pow(x2,2) - 10 * cos(2 * PI * x2) + 10;
    return r;
}
```

### C.2.5. Rosenbrock

```
double FO(double x1, double x2){  
    double r;  
    r = 100*pow( x2-x1*x1 , 2) + pow(x1-1, 2);  
    return r;  
}
```

### C.2.6. Schwefel

```
double FO(double x1, double x2){  
    double r;  
    r = 2 * 418.9829 - x1*sin(sqrt(fabs(x1))) - x2*sin(sqrt(fabs(x2)));  
    return r;  
}
```

### C.3. Técnicas de cruza

A continuación se muestran el código correspondiente a las diferentes técnicas de cruza utilizadas.

#### C.3.1. Cruza de 2 puntos

```
j = 0;
for (i = 0 ; i < nhijos ; i++){
    bitset<CONJ> pA, pB, h1, h2;
    bitset<BITS> s1, s2, s3, s4;
    int p1 = rand() % CONJ;
    int p2 = rand() % CONJ;
    while(p2 == p1)
        p2 = rand() % CONJ;
    if (p2 < p1){
        swap(p1, p2);
    }
    for (k = 0 ; k < BITS ; k++){
        pA[k+BITS] = x[padres[j]][k];
        pA[k] = y[padres[j]][k];
        pB[k+BITS] = x[padres[j+1]][k];
        pB[k] = y[padres[j+1]][k];
    }

    for (k = 0 ; k < p1 ; k++){
        h1[k] = pA[k];
        h2[k] = pB[k];
    }
    for (k = p1 ; k < p2 ; k++){
        h2[k] = pA[k];
        h1[k] = pB[k];
    }
    for (k = p2 ; k < CONJ ; k++){
        h1[k] = pA[k];
        h2[k] = pB[k];
    }

    for (k = 0 ; k < BITS ; k++){
        s1[k] = pA[k+BITS];
        s2[k] = pA[k];
        s3[k] = pB[k+BITS];
        s4[k] = pB[k];
    }
    j += 2;
    newx.push_back(s1);
    newx.push_back(s3);
    newy.push_back(s2);
    newy.push_back(s4);
}
```

## C.3.2. PAX

```

for (i = 0 ; i < CONJ ; i++){
    pint[i] = (pt[i] - 0.5)*(1 - alfa) + 0.5;
    poblp[i] = 0.0;
    pobln[i] = 0.0;
}

for (i = 0 ; i < Tpop ; i++){
    double fc;
    if (fobj[i] >= promedio){
        fc = (fobj[i]-promedio)/sumatoriap;
        for (j = 0 ; j < BITS ; j++){
            int xx = x[i][j];
            int yy = y[i][j];
            poblp[j+BITS] += xx*fc + (xx - 1) * fc;
            poblp[j] += yy*fc + (yy - 1) * fc;
        }
    }
    else{
        fc = (fobj[i]-promedio)/sumatorian;
        for (j = 0 ; j < BITS ; j++){
            int xx = x[i][j];
            int yy = y[i][j];
            pobln[j+BITS] += xx*fc + (xx - 1) * fc;
            pobln[j] += yy*fc + (yy - 1) * fc;
        }
    }
}

for (i = 0 ; i < CONJ ; i++){
    pt[i] = pint[i] + (aprz/2) * (poblp[i] - pobln[i]);
}

j = 0;
double pc1 = double_rand();
double pc2 = double_rand();
for (i = 0 ; i < nhijos ; i++){
    bitset<CONJ> pA, pB, h1, h2;
    bitset<BITS> s1, s2, s3, s4;

    for (k = 0 ; k < BITS ; k++){
        pA[k+BITS] = x[padres[j]][k];
        pA[k] = y[padres[j]][k];
        pB[k+BITS] = x[padres[j+1]][k];
        pB[k] = y[padres[j+1]][k];
    }

    for (k = 0 ; k < CONJ ; k++){
        if (pA[k] == pB[k]){
            h1[k] = pA[k];
        }
    }
}

```

```
        h2[k] = pB[k];
    }
    else{
        if (pt[k] >= pc1)
            h1[k] = 1;
        else
            h1[k] = 0;
        if (pt[k] >= pc2)
            h2[k] = 1;
        else
            h2[k] = 0;
    }
}

for (k = 0 ; k < BITS ; k++){
    s1[k] = pA[k+BITS];
    s2[k] = pA[k];
    s3[k] = pB[k+BITS];
    s4[k] = pB[k];
}
j += 2;
newx.push_back(s1);
newx.push_back(s3);
newy.push_back(s2);
newy.push_back(s4);
}
```

## C.3.3. SANUX

```
//Determinacion de frecuencias y mascara de bits

for (j = 0 ; j < BITS ; j++){
    int countx = 0;
    int county = 0;
    for (i = 0 ; i < Tpop ; i++){
        if (x[i][j] == 1)
            countx++;
        if (y[i][j] == 1)
            county++;
    }
    freqx[j] = (countx * 1.0) / Tpop;
    freqy[j] = (county * 1.0) / Tpop;

    if (freqx[j] <= 0.5){
        psx[j] = freqx[j];
    }
    else{
        psx[j] = 1 - freqx[j];
    }
    if (freqy[j] <= 0.5){
        psy[j] = freqy[j];
    }
    else{
        psy[j] = 1 - freqy[j];
    }
}

double mask = double_rand() / 2;

if (psx[j] > mask )
    mascarax[j] = 0;
else
    mascarax[j] = 1;

if (psy[j] > mask )
    mascaray[j] = 0;
else
    mascaray[j] = 1;
}

j = 0;
for (i = 0 ; i < nhijos ; i++){
    bitset<BITS> s1 ,s2 ,s3 ,s4;

    for (k = 0 ; k < BITS ; k++){
        if (mascarax[k] == 0){
            s1[k] = x[padres[j]][k];
            s3[k] = x[padres[j+1]][k];
        }
    }
}
```

```
    else{
        s1[k] = x[padres[j+1]][k];
        s3[k] = x[padres[j]][k];
    }

    if (mascaray[k] == 0){
        s2[k] = y[padres[j]][k];
        s4[k] = y[padres[j+1]][k];
    }
    else{
        s2[k] = y[padres[j+1]][k];
        s4[k] = y[padres[j]][k];
    }
}
j += 2;
newx.push_back(s1);
newx.push_back(s3);
newy.push_back(s2);
newy.push_back(s4);
}
```

### C.3.4. Selective Crossover

```

j = 0;
for (i = 0 ; i < nhijos ; i++){
    bitset<CONJ> pA, pB, h1, h2;
    bitset<BITS> s1, s2, s3, s4;
    vector<double> domh1(CONJ), domh2(CONJ);
    double fh1, fh2, fpA, fpB;

    for (k = 0 ; k < BITS ; k++){ //Unificación de cadenas
        pA[k+BITS] = x[padres[j]][k];
        pA[k] = y[padres[j]][k];
        pB[k+BITS] = x[padres[j+1]][k];
        pB[k] = y[padres[j+1]][k];
    }

    for (k = 0 ; k < CONJ ; k++){ //Generación de los hijos
        if ( dominancia[padres[j]][k] >= dominancia[padres[j+1]][k] ){
            h1[k] = pA[k];
            h2[k] = pB[k];
            domh1[k] = dominancia[padres[j]][k];
            domh2[k] = dominancia[padres[j+1]][k];
        }
        else{
            h1[k] = pB[k];
            h2[k] = pA[k];
            domh2[k] = dominancia[padres[j]][k];
            domh1[k] = dominancia[padres[j+1]][k];
        }
    }

    for (k = 0 ; k < BITS ; k++){ //Separación de cadenas
        s1[k] = pA[k+BITS];
        s2[k] = pA[k];
        s3[k] = pB[k+BITS];
        s4[k] = pB[k];
    }

    fpA = (mejorindividuo - fobj[padres[j]])/(mejorindividuo -
        peorindividuo);
    fpB = (mejorindividuo - fobj[padres[j+1]])/(mejorindividuo -
        peorindividuo);
    fh1 = ( mejorindividuo - FO(convertir(s1), convertir(s2))
        )/(mejorindividuo - peorindividuo);
    fh2 = ( mejorindividuo - FO(convertir(s3), convertir(s4))
        )/(mejorindividuo - peorindividuo);

    if (fh1 < fpA){
        for (k = 0 ; k < CONJ ; k++)
            if (pA[k] != h1[k]){
                domh1[k] = domh1[k] + (fpA - fh1)*(1-fh1)/Gmax;
            }
    }
}

```



```
    }
  }
  else if ( fh1 < fpB){
    for (k = 0 ; k < CONJ ; k++)
      if (pB[k] != h1[k]){
        domh1[k] = domh1[k] + (fpB - fh1)*(1-fh1)/Gmax;
      }
  }

  if (fh2 < fpA){
    for (k = 0 ; k < CONJ ; k++)
      if (pA[k] != h2[k]){
        domh2[k] = domh2[k] + (fpA - fh2)*(1-fh2)/Gmax;
      }
  }
  else if ( fh2 < fpB){
    for (k = 0 ; k < CONJ ; k++)
      if (pB[k] != h2[k]){
        domh2[k] = domh2[k] + (fpB - fh2)*(1-fh2)/Gmax;
      }
  }

  j += 2;

  newx.push_back(s1);
  newx.push_back(s3);
  newy.push_back(s2);
  newy.push_back(s4);
  newdom.push_back(domh1);
  newdom.push_back(domh2);
}
```

## C.3.5. GPAX

```

//Funciones F1, F2 y F3 utilizadas por la técnica de cruza

double F1(double x, double alfa){
    double r;
    if (alfa <= 0.5){
        r = (pow(x + 2, 2 * alfa + 1))/(2 * alfa + 1);
    }
    else{
        r = (2 * (1 - alfa) * pow(x + 2, (3 - 2 * alfa)/(2 * alfa + 1) )
            )/(3 - 2 * alfa);
    }
    return r;
}

double F2(double x, double beta){
    double r;
    if (beta >= 0.5){
        r = -1 * ( (3*beta*pow(x,3) - 9*beta - 2*pow(x,3) + 4 -
            12*beta*x + 6*x)/(3 * beta) );
    }
    else{
        double c = -1 * sqrt((4 * beta - 2)/(3 * beta - 2));
        if ( -1 < x && x < (-1 * c) ){
            r = ((2-3*beta)*pow(x,3) + 6*(2*beta-1)*x + 9*beta - 4)/(3
                * beta);
        }
        else if ((-1 * c) < x && x < c ){
            r = (27*beta*beta - 30*beta + 8)/(3*beta*(3*beta-2)) + (4 *
                (2*beta-1)*c)/(3*beta);
        }
        else{
            r = ((16*beta-8)*c)/(3*beta)
                + (27*beta*beta - 30*beta + 8)/(3*beta*(3*beta-2))
                + (36*beta*beta*x - 9*pow(x,3)*beta*beta -
                    42*beta*x)/(3*beta*(3*beta-2))
                + (12*beta*pow(x,3) + 12*x -
                    4*pow(x,3))/(3*beta*(3*beta-2));
        }
    }
    return r;
}

double F3(double x, double alfa){
    double r;
    if (alfa <= 0.5){
        r = (1 - pow(2-x, 2*alfa + 1))/(2*alfa + 1);
    }
    else{

```

```

        r = ( (2 - 2*alfa)/(3 - 2*alfa) )*(1 -(2 - pow(x,
            (3-2*alfa)/(2-2*alfa)))));
    }
    return r;
}

//Función de calculo de la x mediante dicotomia

double dicotomia(double y, double alfa , double beta){
int k;
double xmas, xmenos, x;
if ( y < F1(-1, alfa) ){
    xmenos = -2;
    xmas = -1;
    for (k = 0; k < 10 ; k++){
        x = (xmas + xmenos) / 2;
        if( F1(x, alfa) <= y){
            xmenos = x;
        }
        else{
            xmas = x;
        }
    }
}
else if (y <= (F2(1, beta) + F1(-1, alfa))){
    xmenos = -1;
    xmas = 1;
    for (k = 0; k < 10 ; k++){
        x = (xmas + xmenos) / 2;
        if( F2(x, beta) <= y){
            xmenos = x;
        }
        else{
            xmas = x;
        }
    }
}
else{
    xmenos = 1;
    xmas = 2;
    for (k = 0; k < 10 ; k++){
        x = (xmas + xmenos) / 2;
        if( F3(x, alfa) <= y){
            xmenos = x;
        }
        else{
            xmas = x;
        }
    }
}
x = (xmas + xmenos) / 2;

```

```
return x;
}

/*Proceso de recombinación GPAX*/

j = supervivientes;
for ( i = 0 ; i < nhijos ; i+=2){
    p1j = x1[padres[i]];
    p2j = x1[padres[i+2]];
    y = double_rand();
    if (p1j > p2j)
        swap(p1j,p2j);
    seleccionx1[j] = p1j + dicotomia(y, alfa , beta) * (p2j - p1j) / 2;

    p1j = x2[padres[i]];
    p2j = x2[padres[i+2]];
    y = double_rand();
    if (p1j > p2j)
        swap(p1j,p2j);
    seleccionx2[j] = p1j + dicotomia(y, alfa , beta) * (p2j - p1j) / 2;
    j++;

    if (FO(seleccionx1[j], seleccionx2[j]) < best){
        bestalfa = alfa;
        bestbeta = beta;
    }

    //Perturbación alfa y beta

    if (double_rand() >= 0.5)
        alfa +=0.1;
    else
        alfa -=0.1;

    if (double_rand() >= 0.5)
        beta -=0.1;
    else
        beta +=0.1;

    if (alfa > 1)
        alfa = 1;
    if (alfa < 0)
        alfa = 0;
    if (beta > 1)
        beta = 1;
    if (beta < 0)
        beta = 0;

    //_____
```

```
p1j = x1[padres[i]];
p2j = x1[padres[i+2]];
y = double_rand();
if (p1j > p2j)
    swap(p1j,p2j);
seleccionx1[j] = p1j + dicotomia(y, alfa , beta) * (p2j - p1j) / 2;

p1j = x2[padres[i]];
p2j = x2[padres[i+2]];
y = double_rand();
if (p1j > p2j)
    swap(p1j,p2j);
seleccionx2[j] = p1j + dicotomia(y, alfa , beta) * (p2j - p1j) / 2;
j++;

if (FO(seleccionx1[j], seleccionx2[j]) < best){
    bestalfa = alfa;
    bestbeta = beta;
}
}
```

### C.3.6. Recombinación intermedia

```
j = supervivientes;
for ( i = 0 ; i < nhijos ; i += 2){
    p1 = padres[i];
    p2 = padres[i+1];
    a = double_rand();
    seleccionx1[j] =x1[p2]*a + x1[p1]*(1-a);
    seleccionx2[j] =x2[p2]*a + x2[p1]*(1-a);
    j++;
    if (j < Tpop){
        seleccionx1[j] =x1[p1]*a + x1[p2]*(1-a);
        seleccionx2[j] =x2[p1]*a + x2[p2]*(1-a);
    }
    j++;
}
```

### C.3.7. SBX

```
j = supervivientes;
for ( i = 0 ; i < nhijos ; i += 2){
    p1 = padres[i];
    p2 = padres[i+1];

    a = double_rand();
    if ( a <= 0.5)
        beta = pow(2*a, 1.0/3.0); //Se toma el parámetro nc como 2 por
        recomendación del autor
    else
        beta = pow(1/(2*(1-a)), 1.0/3.0);

    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]-beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]-beta*fabs(x2[p2]-x2[p1]));
    j++;
    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]+beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]+beta*fabs(x2[p2]-x2[p1]));
    j++;
}
```

### C.3.8. SBX Adaptativo

```

j = supervivientes;
for ( i = 0 ; i < nhijos ; i += 2){
    double nc1, nc2, bt, c1, c2, pp1, pp2;
    p1 = padres[i];
    p2 = padres[i+1];

    pp1 = FO(x1[p1], x2[p1]);
    pp2 = FO(x1[p2], x2[p2]);

    a = double_rand();
    if ( a <= 0.5)
        beta = pow(2*a, 1.0/(1 + nc));
    else
        beta = pow(1/(2*(1-a)), 1.0/(1 + nc));

    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]-beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]-beta*fabs(x2[p2]-x2[p1]));
    c1 = FO(seleccionx1[j], seleccionx2[j]);
    j++;
    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]+beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]+beta*fabs(x2[p2]-x2[p1]));
    c2 = FO(seleccionx1[j], seleccionx2[j]);
    j++;

    if (pp1 != pp2){
        if (pp1 > pp2)
            swap(pp1, pp2);

        bt = 1 + (2 * (c2 - pp2))/(pp2 - pp1);
        if (bt > 1){
            if (c2 < pp2 || c1 < pp1){
                nc2 = ((nc + 1) * log(bt))/log(1 + alfa * (bt - 1)) - 1;
                nc1 = nc2;
            }
            if (c2 > pp1 && c2 > pp2)
                nc2 = ((nc + 1) * log(bt))/log(1 + (bt - 1) / alfa) - 1;
            if (c1 > pp1 && c1 > pp2)
                nc1 = ((nc + 1) * log(bt))/log(1 + (bt - 1) / alfa) - 1;
        }
    }
    else if (bt < 1){
        if (c2 > pp1 && c2 > pp2)
            nc2 = (1 + nc) / alfa - 1;
        else
            nc2 = alfa * (1 + nc) - 1;

        if (c1 > pp1 && c1 > pp2)
            nc1 = (1 + nc) / alfa - 1;
        else
    }

```

```
        nc1 = alfa * (1 + nc) - 1;
    }
    if (c2 < c1)
        nc = nc2;
    if (c1 < c2)
        nc = nc1;

    if (nc < 0)
        nc = 0;
    if (nc > 50)
        nc = 50;
}
}
```



## C.3.9. XHC

```

j = supervivientes;
for ( i = 0 ; i < nhijos ; i += 2){
    double pA_1, pA_2, pB_1, pB_2;

    pA_1 = x1[padres[i]];
    pA_2 = x2[padres[i]];
    pB_1 = x1[padres[i+1]];
    pB_2 = x2[padres[i+1]];

    for (int nt = 0 ; nt < 10 ; nt++){
        vector<double> noffx, noffy;
        int mejornoff;
        double bestnoff;

        for (int noff = 0 ; noff < 5 ; noff++){ //Generación de la
            descendencia
                double I, l1, u1, l2, u2, z1, z2;

                I = fabs(pA_1 - pB_1);
                l1 = max(LI, pA_1 - I * alfa);
                u1 = min(LS, pA_1 + I * alfa);
                l2 = max(LI, pB_1 - I * alfa);
                u2 = min(LS, pB_1 + I * alfa);
                z1 = u1 - double_rand() * (u1 - l1);
                z2 = u2 - double_rand() * (u2 - l2);
                noffx.push_back(z1);
                noffx.push_back(z2);

                I = fabs(pA_2 - pB_2);
                l1 = max(LI, pA_2 - I * alfa);
                u1 = min(LS, pA_2 + I * alfa);
                l2 = max(LI, pB_2 - I * alfa);
                u2 = min(LS, pB_2 + I * alfa);
                z1 = u1 - double_rand() * (u1 - l1);
                z2 = u2 - double_rand() * (u2 - l2);
                noffy.push_back(z1);
                noffy.push_back(z2);
            }

            mejornoff = 0; //Búsqueda del mejor hijo generado
            bestnoff = FO(noffx[0], noffy[0]);
            for (int k = 1 ; k < noffx.size(); k++){
                if ( FO(noffx[k], noffy[k]) < bestnoff){
                    mejornoff = k;
                    bestnoff = FO(noffx[k], noffy[k]);
                }
            }

            if ( FO(pA_1, pA_2) < FO(pB_1, pB_2) ){
    
```

```
        if ( bestnoff < FO(pB_1, pB_2) ){
            pB_1 = noffx[mejornoff];
            pB_2 = noffy[mejornoff];
        }
    }
    else{
        if ( bestnoff < FO(pA_1, pA_2) ){
            pA_1 = noffx[mejornoff];
            pA_2 = noffy[mejornoff];
        }
    }
}
seleccionx1[j] = pA_1;
seleccionx2[j] = pA_2;
j++;
seleccionx1[j] = pB_1;
seleccionx2[j] = pB_2;
j++;
}
```

## C.4. Ejemplo completo

En este apartado se presentan dos códigos fuentes completos, uno con codificación binaria y uno con codificación real.

### C.4.1. Función de Rastringin y PAX

```
/*
ARCHIVO:      Rastringin_PAX.cpp

DESCRIPCIÓN:  Contiene la implementación del algoritmo genético con
               codificación
               binaria que busca la solución de la función Rastringin con dos variables. Se
               emplea la técnica de cruza PAX.
*/

//Inclusión de
//librerías-----

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <bitset>
#include <vector>
#include <iostream>
#include <algorithm>
#include <deque>

using namespace std;

//Declaración de macros

#define BITS 24      // Se considera un rango de -30 a 30 con 6 decimales de
                    // precisión
#define LI -5.12    // Limite inferior del rango
#define LS 5.12     // Limite superior del rango
#define CONJ 48     // Largo de la cadena juntando las variables
#define PI 3.1416

/*Declaración de variables globales*/
//Vectores de bitset's que contienen la población y la nueva población
vector<bitset<BITS>> x, newx;
vector<bitset<BITS>> y, newy;

//Vector que contienen a los indices de los padres
deque<int> padres;

//Vector que almacena el resultado de la función objetivo para cada
//individuo de la población
vector<double> fobj;
```

```

//Implementación de la funcion objetivo

double FO(double x1, double x2){
    double r;
    r = pow(x1,2) - 10 * cos(2 * PI * x1) + 10;
    r += pow(x2,2) - 10 * cos(2 * PI * x2) + 10;
    return r;
}

//Función que realiza el mapeo Código gray-Binario-Entero-Real

double convertir(bitset<BITS> origen){
    //Paso 1: Código de Gray a binario
    bitset<BITS> binario;
    bitset<1> valor;

    valor[0] = origen[BITS-1];
    binario[BITS-1] = valor[0];
    for (int k = BITS-2 ; k >= 0 ; k--){
        if (origen[k] == 1){
            valor.flip();
        }
        binario[k] = valor[0];
    }
    //Paso 2: Binario a entero
    long entero = binario.to_ulong();

    //Paso 3: Entero a real
    double x;

    x = LI + ((LS - LI) * entero)/(pow(2, BITS) - 1);

    return x;
}

//Función que retorna un numero aleatorio entre 0 y 1
double double_rand(){
    return (rand() / (double)RAND_MAX);
}

//Función principal
main-----

int main(int argc, char *argv[])
{
    //Parámetros del algoritmo genético
    int semilla; //Semilla para la generación de números aleatorios
    int Tpop; //Tamaño de la población
    int Gmax; //Número de generaciones

```

```

double Ts;    //Tasa de supervivencia
double Tc;    //Tasa de recombinación
double Tm;    //Tasa de mutación

/*Declaración de variables auxiliares*/
int t = 0;    //Contador de generaciones
int i, j, k;  //Variables indice
bitset<BITS> auxiliar;    //Bitset auxiliar

int nsel;    //Número de supervivientes
int npadres; //Número de padres
int nhijos;  //Número de hijos

double peorindividuo; //Peor valor de la función objetivo en la
    población
double mejorindividuo; //Mejor valor de la función objetivo en la
    población
double promedio;    //Valor promedio de la población
double sumafo;    //Suma de los valores de la función objetivo de
    la población
int indicepeor;    //Indice del peor individuo
int indicemejor;  //Indice del mejor individuo
double r;    //Variable auxiliar

vector<double> apt;    //Vector de aptitudes , se usa en la selección
vector<double> aptrel; //Vector de aptitudes relativas , se usa en la
    selección
double apttotal;    //Suma de las aptitudes , se usa en la selección
double ptr;    //Variable aleatoria entre 0 y 1, se usa en la
    selección
double suma;    //Variable auxiliar , se usa en la selección

double mutador;    //Variable auxiliar en el proceso de mutación

double sumatoriap; //Sumatoria positiva
double sumatorian; //Sumatoria negativa
vector<double> pt(CONJ); //Vector de probabilidades
vector<double> pint(CONJ); //Vector de probabilidades intermedias
double alfa;    //Factor de olvido
double aprz;    //Tasa de aprendizaje
vector<double> poblp(CONJ); //Sumatoria de ejemplos positivos
vector<double> pobln(CONJ); //Sumatoria de ejemplos negativos

/*Inicialización de parámetros*/
semilla = atoi(argv[1]);
Tpop = atoi(argv[2]);
Gmax = atoi(argv[3]);

Ts = atof(argv[4]);
Tc = 1 - Ts;
    
```

```
Tm = atof(argv[5]);

nssel = Tpop * Ts;
npadres = 2 * Tpop * Tc;
nhijos = Tpop * Tc;

/*Inicialización de la semilla de números aleatorios*/
srand(semilla);

//Archivo para guardar los datos
FILE *data, *registro;
data = fopen(argv[6], "w");
registro = fopen("resultados.txt", "a");

/*Generación de la población inicial.
   Se genera un número aleatorio entre 0 y 100. Dependiendo si el valor
   es menor o mayor a 50 se le da valor al bit
*/
for (i = 0 ; i < Tpop ; i++){
    for (j = 0 ; j < BITS ; j++){
        if ((rand() % 100) < 50)
            auxiliar[j] = 0;
        else
            auxiliar[j] = 1;
    }
    x.push_back(auxiliar);
}
for (i = 0 ; i < Tpop ; i++){
    for (j = 0 ; j < BITS ; j++){
        if ((rand() % 100) < 50)
            auxiliar[j] = 1;
        else
            auxiliar[j] = 0;
    }
    y.push_back(auxiliar);
}

//Calculo del valor de la función objetivo para cada individuo de la
población
r = FO(convertir(x[0]), convertir(y[0]));
sumafo = r;
peorindividuo = r;
mejorindividuo = r;
indicepeor = 0;
indicemejor = 0;
fobj.push_back(r);
for (i = 1 ; i < Tpop ; i++){
    r = FO(convertir(x[i]), convertir(y[i]));
    if (r < mejorindividuo){
        mejorindividuo = r;
        indicemejor = i;
    }
}
```

```

    }
    if (r > peorindividuo){
        peorindividuo = r;
        indicepeor = i;
    }
    fobj.push_back(r);
    suma = suma + r;
}
promedio = sumafo / Tpop;

//Calculo de las sumatorias
sumatorian = 0.0;
sumatoriap = 0.0;
for (i = 0 ; i < Tpop ; i++){
    if (fobj[i] >= promedio)
        sumatoriap = sumatoriap + fobj[i] - promedio;
    else
        sumatorian = sumatorian + fobj[i] - promedio;
}

//Vector inicial de probabilidades

for (i = 0 ; i < CONJ ; i++){
    pt[i] = double_rand();
}
alfa = 0.3;
aprz = 0.2;

fprintf(data , "%20.6f_%20.6f_%20.6f_%20.6f_%20.6f\n" ,
        convertir(x[indicemejor]),convertir(y[indicemejor]), mejorindividuo ,
        peorindividuo , promedio);

//Ciclo evolutivo
while (t < Gmax){

    //Selección de supervivientes (SUS)
    apttotal = 0.0;
    for ( i = 0 ; i < Tpop ; i++){
        double dif = peorindividuo - fobj[i];
        apt.push_back(dif);
        apttotal = apttotal + dif;
    }
    for ( i = 0 ; i < Tpop ; i++){
        double rel = apt[i] / apttotal;
        aptrel.push_back(rel);
    }
    ptr = double_rand() / nsel;
    suma = 0.0;
    i = 0;
    j = 0;
    while (i < nsel){

```

```

suma = suma + aptrel[j];
while (suma > ptr){
    newx.push_back(x[j]);
    newy.push_back(y[j]);
    i++;
    ptr = ptr + 1.0/nsel;
}
j++;
}

//Selección de padres
ptr = double_rand() / npadres;
suma = 0.0;
i = 0;
j = 0;
while (i < npadres){
    suma = suma + aptrel[j];
    while (suma > ptr){
        if (padres.empty())
            padres.push_back(j);
        else if (padres.back() == j)
            padres.push_front(j);
        else
            padres.push_back(j);
        i++;
        ptr = ptr + 1.0/npadres;
    }
    j++;
}

//PAX

for (i = 0 ; i < CONJ ; i++){
    pint[i] = (pt[i] - 0.5)*(1 - alfa) + 0.5;
    poblp[i] = 0.0;
    poblm[i] = 0.0;
}

for (i = 0 ; i < Tpop ; i++){
    double fc;
    if (fobj[i] >= promedio){
        fc = (fobj[i]-promedio)/sumatoriap;
        for (j = 0 ; j < BITS ; j++){
            int xx = x[i][j];
            int yy = y[i][j];
            poblp[j+BITS] += xx*fc + (xx - 1) * fc;
            poblm[j] += yy*fc + (yy - 1) * fc;
        }
    }
    else{
        fc = (fobj[i]-promedio)/sumatorian;
    }
}

```



```

        for (j = 0 ; j < BITS ; j++){
            int xx = x[i][j];
            int yy = y[i][j];
            pobl[n[j+BITS]] += xx*fc + (xx - 1) * fc;
            pobl[n[j]] += yy*fc + (yy - 1) * fc;
        }
    }

    for (i = 0 ; i < CONJ ; i++){
        pt[i] = pint[i] + (aprz/2) * (pobl[p[i]] - pobl[n[i]]);
    }

    j = 0;
    double pc1 = double_rand();
    double pc2 = double_rand();
    for (i = 0 ; i < nhijos ; i++){
        bitset<CONJ> pA, pB, h1, h2;
        bitset<BITS> s1, s2, s3, s4;

        for (k = 0 ; k < BITS ; k++){
            pA[k+BITS] = x[padres[j]][k];
            pA[k] = y[padres[j]][k];
            pB[k+BITS] = x[padres[j+1]][k];
            pB[k] = y[padres[j+1]][k];
        }
        /**

        for (k = 0 ; k < CONJ ; k++){
            if (pA[k] == pB[k]){
                h1[k] = pA[k];
                h2[k] = pB[k];
            }
            else{
                if (pt[k] >= pc1)
                    h1[k] = 1;
                else
                    h1[k] = 0;
                if (pt[k] >= pc2)
                    h2[k] = 1;
                else
                    h2[k] = 0;
            }
        }

        /**
        for (k = 0 ; k < BITS ; k++){
            s1[k] = pA[k+BITS];
            s2[k] = pA[k];
            s3[k] = pB[k+BITS];
            s4[k] = pB[k];
    
```

```
    }
    j += 2;
    newx.push_back(s1);
    newx.push_back(s3);
    newy.push_back(s2);
    newy.push_back(s4);
}

//Mutación
for (i = 0 ; i < Tpop ; i++){
    for (j = 0 ; j < BITS ; j++){
        mutador = double_rand();
        if (mutador < Tm)
            newx[i].flip(j);
        mutador = double_rand();
        if (mutador < Tm)
            newy[i].flip(j);
    }
}

//Elitismo

r = FO(convertir(newx[0]), convertir(newy[0]));
peorindividuo = r;
indicepeor = 0;
for (i = 1 ; i < Tpop ; i++){
    r = FO(convertir(newx[i]), convertir(newy[i]));
    if (r > peorindividuo){
        peorindividuo = r;
        indicepeor = i;
    }
}
newx[indicepeor] = x[indicemejor];
newy[indicepeor] = y[indicemejor];

//Actualización de la población
x = newx;
y = newy;

//Calculo del valor de la función objetivo para cada individuo de la
nueva población
fobj.clear();
r = FO(convertir(x[0]), convertir(y[0]));
sumafo = r;
peorindividuo = r;
mejorindividuo = r;
indicepeor = 0;
indicemejor = 0;
fobj.push_back(r);
for (i = 1 ; i < Tpop ; i++){
    r = FO(convertir(x[i]), convertir(y[i]));
```

```

        if (r < mejorindividuo){
            mejorindividuo = r;
            indicemejor = i;
        }
        if (r > peorindividuo){
            peorindividuo = r;
            indicepeor = i;
        }
        fobj.push_back(r);
        suma = suma + r;
    }
    promedio = sumafo / Tpop;
    //Calculo de las sumatorias
    sumatorian = 0.0;
    sumatoriap = 0.0;
    for (i = 0 ; i < Tpop ; i++){
        if (fobj[i] >= promedio)
            sumatoriap = sumatoriap + fobj[i] - promedio;
        else
            sumatorian = sumatorian + fobj[i] - promedio;
    }

    fprintf(data , "%20.6f_%20.6f_%20.6f_%20.6f_%20.6f\n" ,
        convertir(x[indicemejor]),convertir(y[indicemejor]), mejorindividuo ,
        peorindividuo , promedio);

    //Incremento en el contador de generaciones
    t++;

    //Reinicio de vectores auxiliares
    apt.clear();
    aptrel.clear();
    padres.clear();
    newx.clear();
    newy.clear();
}
fprintf(registro , "%5d_%5d_%5d_%4.2f_%4.2f_%20.6g_%20.6g_%20.6g_%20.6g_
%20.6g\n" , semilla , Tpop , Gmax , Ts , Tm,
    convertir(x[indicemejor]),convertir(y[indicemejor]), mejorindividuo ,
    peorindividuo , promedio);
fclose(data);
fclose(registro);
return 0;
}

```

## C.4.2. Función de Griewank y SBX Adaptativo

```
/*
ARCHIVO:      Griewank_SBXA.cpp

DESCRIPCIÓN:  Contiene la implementación del algoritmo genético con
               codificación
               real que busca la solución de la función de Griewank con dos variables. Se
               emplea la técnica de cruza SBX Adaptativa.

*/

//Inclusión de librerías
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <cstdio>
#include <vector>
#include <deque>

using namespace std;

//Declaración de macros
#define LI -600.0 // Limite inferior del rango
#define LS 600   // Limite superior del rango

//Implementación de la función objetivo
double FO(double x1, double x2){
    double r;
    double s = (pow(x1, 2) + pow(x2, 2))/4000;
    double p = cos(x1) * cos(x2/sqrt(2));
    r = s - p +1;
    return r;
}

//Función que retorna un numero aleatorio entre 0 y 1
double double_rand(){
    return (rand() / (double)RAND_MAX);
}

//Función principal Main
int main(int argc, char *argv[]){

    //Declaracion de Parámetros
    int t;          //Contador de Generaciones
```

```

int Tpop;      //Tamaño de la población
int Gmax;      //Número maximo de generaciones
double Ts;     //Tasa de supervivencia (20% al 80%)
double Tc;     //Tasa de recombinación (1 – Ts)
double Tm;     //Tasa de mutación (1% al 20%)
int semilla;  //Semilla para los números aleatorios

//Inicialización de Parámetros
semilla = atoi(argv[1]);
Tpop = atoi(argv[2]);
Gmax = atoi(argv[3]);
Ts = atof(argv[4]);
Tc = 1– Ts;
Tm = atof(argv[5]);

//Variables auxiliares para el proceso

vector<double> apt(Tpop);           //Vector de aptitudes
vector<double> aptrel(Tpop);       //Vector de aptitudes relativas
vector<double> seleccionx1(Tpop);  //Vector de los supervivientes de
    la seleccion
vector<double> seleccionx2(Tpop);  //Vector de los supervivientes de
    la seleccion
double sum;                       //Sumatoria de las aptitudes
double rs;                         //Número aleatorio entre 0 y 1
int supervivientes;               //Número de sobrevivientes
int i, j;                          //Indices

double a;                          //Parametro de recombinación
double beta;                       //Parametro de recombinación
int p1, p2;                        //Indice de los padres que
    generan la recombinación

int mejor;                          //Indice de la mejor solución de la
    población inicial
int peor;                            //Indice de la peor solución de la
    poblacion generada
double promedio;                   //Promedio de las soluciones
double apttotal;                   //Suma de las aptitudes
double ptr;
int npadres;                       //Número de padres a escoger
int nhijos;                        //Número de hijos a generar

int nmut;                           //Numero de mutaciones
int u;                              //Parametro para la mutación
int nm;                             //Indice de distribución
double delta;                      //Rango de la variable
double sigma;
int ms;
    
```

```

deque<int> padres; //Doble cola que contienen a los
                  indices de los padres

double nc; //Parámetros para la cruza
double alfa;

//Inicialización de la Semilla de Números Aleatorios
srand (semilla);

//Inicialización de la población

vector<double> x1(Tpop); //Individuos de la población
vector<double> x2(Tpop); //Individuos de la población
vector<double> r(Tpop); //Resultados de la evaluación en la FO

for (i = 0 ; i < Tpop ; i++) {
    x1[i] = 5.12 - double_rand() * 10.24;
    x2[i] = 5.12 - double_rand() * 10.24;
    r[i] = FO(x1[i], x2[i]);
}

mejor = 0; //Busqueda de la mejor solución en la poblacion actual
promedio = 0.0;
for (i = 1 ; i < Tpop ; i++){
    if (r[i] < r[mejor])
        mejor = i;
    promedio = promedio+r[i];
}
promedio = promedio/Tpop;

peor = 0; //Busqueda de la peor solución en la nueva población
for (i = 1 ; i < Tpop ; i++){
    if (r[i] > r[peor])
        peor = i;
}
FILE *evolucion;
evolucion = fopen(argv[6], "w");
fprintf(evolucion, "%20.6g_%20.6g_%20.6g_%20.6g_%20.6g\n", x1[mejor],
        x2[mejor], r[mejor], r[peor], promedio);

//Inicio de las iteraciones

t = 0;
supervivientes = Tpop * Ts;
npadres = 2 * Tpop * Tc;
nhijos = Tpop * Tc;

nc = 2;
alfa = 1.5;
    
```

```
while (t < Gmax){

    /*Selección de supervivientes SUS*/
    apttotal = 0.0;
    for (i = 0 ; i < Tpop ; i++){
        apt[i] = r[peor] - r[i];
        apttotal = apttotal + apt[i];
    }
    for (i = 0; i < Tpop; ++i)
    {
        aptrel[i] = apt[i] / apttotal;
    }
    ptr = double_rand() / supervivientes;

    i = 0;
    j = 0;
    double suma = 0.0;
    while (i < supervivientes){
        suma = suma + aptrel[j];
        while(suma > ptr){
            seleccionx1[i] = x1[j];
            seleccionx2[i] = x2[j];
            i++;
            ptr = ptr + 1.0/supervivientes;
        }
        j++;
    }

    /*Selección de padres*/
    ptr = double_rand() / npadres;
    suma = 0.0;
    i = 0;
    j = 0;
    while (i < npadres){
        suma = suma + aptrel[j];
        while(suma > ptr){
            if (padres.empty())
                padres.push_back(j);
            else if (padres.back() == j)
                padres.push_front(j);
            else
                padres.push_back(j);
            i++;
            ptr = ptr + 1.0/npadres;
        }
        j++;
    }

    /*Proceso de recombinación*/
    j = supervivientes;
```

```

for ( i = 0 ; i < nhijos ; i += 2){
    double nc1, nc2, bt, c1, c2, pp1, pp2;
    p1 = padres[i];
    p2 = padres[i+1];

    pp1 = FO(x1[p1], x2[p1]);
    pp2 = FO(x1[p2], x2[p2]);

    a = double_rand();
    if ( a <= 0.5)
        beta = pow(2*a, 1.0/(1 + nc));
    else
        beta = pow(1/(2*(1-a)), 1.0/(1 + nc));

    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]-beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]-beta*fabs(x2[p2]-x2[p1]));
    c1 = FO(seleccionx1[j], seleccionx2[j]);
    j++;
    seleccionx1[j] = 0.5*(x1[p1]+x1[p2]+beta*fabs(x1[p2]-x1[p1]));
    seleccionx2[j] = 0.5*(x2[p1]+x2[p2]+beta*fabs(x2[p2]-x2[p1]));
    c2 = FO(seleccionx1[j], seleccionx2[j]);
    j++;

    if (pp1 != pp2){
        if (pp1 > pp2)
            swap(pp1, pp2);

        bt = 1 + (2 * (c2- pp2))/(pp2 - pp1);
        if (bt > 1){
            if (c2 < pp2 || c1 < pp1){
                nc2 = ((nc + 1) * log(bt))/log(1 + alfa * (bt - 1) ) -
                    1;
                nc1 = nc2;
            }
            if (c2 > pp1 && c2 > pp2)
                nc2 = ((nc + 1) * log(bt))/log(1 + (bt - 1) / alfa ) -
                    1;
            if (c1 > pp1 && c1 > pp2)
                nc1 = ((nc + 1) * log(bt))/log(1 + (bt - 1) / alfa ) -
                    1;
        }
        else if (bt < 1){
            if (c2 > pp1 && c2 > pp2)
                nc2 = (1 + nc) / alfa - 1;
            else
                nc2 = alfa * (1 + nc) - 1;

            if (c1 > pp1 && c1 > pp2)
                nc1 = (1 + nc) / alfa - 1;
            else

```



```

        nc1 = alfa * (1 + nc) - 1;

    }

    if (c2 < c1)
        nc = nc2;
    if (c1 < c2)
        nc = nc1;

    if (nc < 0)
        nc = 0;
    if (nc > 50)
        nc = 50;
}

/* Mutación */
nm = 100 + t;
nmut = Tm * Tpop;
delta = LS - LI;
for (i = 0 ; i < nmut ; i++){
    u = double_rand();

    if (u <= 0.5)
        sigma = pow((2*u) ,1/(nm+1));
    else
        sigma = 1 - pow((2*(1-u)) ,1/(nm+1));

    ms = rand() % Tpop;

    seleccionx1[ms] = seleccionx1[ms]+delta*sigma;
    seleccionx2[ms] = seleccionx2[ms]+delta*sigma;
}

/* Elitismo */
mejor = 0; //Busqueda de la mejor solución en la población actual
for (i = 1 ; i < Tpop ; i++){
    if (r[i] < r[mejor])
        mejor = i;
}

for (i = 0 ; i < Tpop ; i++){ //Evaluación de la nueva población
    r[i] = FO(seleccionx1[i], seleccionx2[i]);
}

peor = 0; //Busqueda de la peor solución en la nueva población
for (i = 1 ; i < Tpop ; i++){
    if (r[i] > r[peor])
        peor = i;
}

```

```

seleccionx1[peor] = x1[mejor];          //Sustitución de los datos
seleccionx2[peor] = x2[mejor];
r[peor] = FO(x1[mejor], x2[mejor]);
x1 = seleccionx1;
x2 = seleccionx2;

//Datos para el archivo evolución: Mejor, Peor, Promedio
mejor = 0;          //Busqueda de la mejor solución en la población actual
promedio = 0.0;
for (i = 1 ; i < Tpop ; i++){
    if (r[i] < r[mejor])
        mejor = i;
    promedio = promedio+r[i];
}
promedio = promedio/Tpop;

peor = 0;          //Busqueda de la peor solución en la nueva población
for (i = 1 ; i < Tpop ; i++){
    if (r[i] > r[peor])
        peor = i;
}

fprintf(evolucion , "%20.6g_%20.6g_%20.6g_%20.6g_%20.6g\n", x1[mejor],
        x2[mejor], r[mejor], r[peor], promedio);

t++;

//Reinicio de vectores auxiliares
padres.clear();
}

/*Busqueda de la mejor solución en la ultima población*/

mejor = 0;
for (i = 1 ; i < Tpop ; i++){
    if (r[i] < r[mejor])
        mejor = i;
}

FILE *pruebas;

pruebas = fopen("resultados.txt", "a");
fprintf(pruebas , "%5d_%5d_%5d_%4.2f_%4.2f_%15.6g_%15.6g_%15.6g_%30.6g_
%30.6g\n", Tpop, Gmax, semilla, Ts, Tm, x1[mejor], x2[mejor],
        r[mejor], r[peor], promedio);
fclose(pruebas);
fclose(evolucion);
return 0;
}

```