

UNIVERSIDAD INDUSTRIAL DE SANTANDER

OPTIMIZACIÓN DEL PESO DE
CERCHAS EN 3D MEDIANTE
ALGORITMOS GENÉTICOS
(GA) Y OPTIMIZACIÓN POR
ENJAMBRE DE PARTÍCULAS
(PSO)

TRABAJO DE GRADO MODALIDAD INVESTIGACIÓN
PRESENTADO COMO REQUISITO PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL

JORGE ALBERTO MOLINA MOJICA
EDUARDO ALFONSO PÁEZ ESTEBAN



2010

ESCUELA DE INGENIERÍA CIVIL
FACULTAD DE INGENIERÍAS FÍSICOMECAÑICAS

**OPTIMIZACIÓN DEL PESO DE CERCHAS EN 3D MEDIANTE ALGORITMOS
GENÉTICOS (GA) Y OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)**

JORGE ALBERTO MOLINA MOJICA

EDUARDO ALFONSO PÁEZ ESTEBAN

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA CIVIL**

2010

**OPTIMIZACIÓN DEL PESO DE CERCHAS EN 3D MEDIANTE ALGORITMOS
GENÉTICOS (GA) Y OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)**

JORGE ALBERTO MOLINA MOJICA

EDUARDO ALFONSO PÁEZ ESTEBAN

TRABAJO DE GRADO MODALIDAD INVESTIGACIÓN PRESENTADO COMO
REQUISITO PARA OPTAR AL TÍTULO DE:

INGENIERO CIVIL

DIRECTOR:

PHD. OSCAR J. BEGAMBRE C.

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
BUCARAMANGA**

2010

AGRADECIMIENTOS

A nuestra Alma Mater y en especial a la Escuela de Ingeniería Civil, la cual por medio de su profesorado nos formo hasta alcanzar este logro.

Al Ing. Civil Oscar Javier Begambre Carrillo por el apoyo, guía, comprensión y gran dedicación brindada durante el desarrollo de este proyecto de investigación.

DEDICATORIA

A Dios por darme la sabiduría, fortaleza y la virtud para culminar con éxito esta etapa de mi vida.

A mis padres, Valentín y Beatriz, por brindarme su apoyo incondicional y por enseñarme los valores que me llevan por un buen camino en la vida.

A mis hermanos, por ser mis guías y mis mayores ejemplos, por los consejos y conocimientos que me han regalado con su amistad.

A Leyda por el amor y el afecto, comprensión, consejos, apoyo y estímulo que me brindó para cumplir con este logro.

A Eduardo por ser el mejor complemento en este logro.

A mis amigos por la compañía, comprensión y ánimo que me dieron día tras día.

Jorge Alberto Molina Mojica

DEDICATORIA

A Dios por regalarme la sabiduría, fuerza y voluntad para lograr esta meta.

A mis padres, Eduardo e Isabel, por ser los guías en mi camino de vida.

*A mi hermana, Nathalia, por ser el mejor ejemplo a seguir,
por los consejos y conocimientos que me ha regalado con su amistad.*

*A Cindy por el amor, cariño, comprensión, apoyo y estímulo
que me brindo para cumplir con este logro.*

A Jorge por ser el mejor complemento en este logro.

*A mis amigos por la compañía, comprensión
y ánimo que me dieron día tras día.*

Eduardo Páez Esteban

TABLA DE CONTENIDO

INTRODUCCIÓN	26
OBJETIVOS.....	29
1. GENERALIDADES Y ENFOQUE PRECEDENTE	30
2. LA OPTIMIZACIÓN	33
2.1 GENERALIDADES	33
2.2 OPTIMIZACIÓN CLÁSICA.....	34
2.2.1 Método de Newton	35
2.2.2 El método simplex.....	37
2.2.3 Otros métodos.....	38
2.3 OPTIMIZACIÓN HEURÍSTICA	38
2.3.1 Algoritmos de optimización	40
2.3.2 Optimización por enjambre de partículas (PSO)	41
2.3.3 Algoritmos genéticos (GA)	52
2.3.4 Otras metodologías	71
3. INVESTIGACIÓN	75
3.1 DISEÑO	75
3.1.1 Definición de cargas.....	75
3.1.2 Resistencia de diseño a tensión.....	76
3.1.3 Resistencia de diseño a compresión.....	77
3.1.4 Datos de la estructura	78
3.1.5 Modelo matemático para problemas de optimización estructural con variables discretas.....	87
3.2 OPTIMIZACIÓN.....	88
3.2.1 Optimización del peso de una cercha 3d mediante los algoritmos PSO y GA	88
3.2.2 Descripción del algoritmo “pso” implementado en el programa Matlab® para la optimización del peso de la cercha.	92

3.2.3	Descripción del algoritmo genético “GA” implementado en el programa Matlab® para la optimización del peso de la cercha.	97
4.	RESULTADOS NUMÉRICOS	101
4.1	RESULTADOS DE LA OPTIMIZACIÓN DEL PESO DE LA CERCHA EN ESTUDIO	101
4.2	RESULTADOS DE LA OPTIMIZACIÓN MEDIANTE OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS	102
4.2.1	Resultados para el caso 1	102
4.2.2	Resultados para el caso 2	115
4.3	RESULTADOS DE LA OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS	135
4.3.1	Resultados para el caso 1	135
4.3.2	Resultados para el caso 2	155
4.4	PRESUPUESTO PARA COSTOS DEL MATERIAL	175
5.	CONCLUSIONES Y RECOMENDACIONES	179
	BIBLIOGRAFÍA.....	184

LISTA DE FIGURAS

Figura 1. Clasificación de los métodos de optimización más relevantes. Por uniformidad con la literatura científica se conserva la denominación original en inglés de cada método.	31
Figura 2. Modelado del PSO utilizando como símil el movimiento de un enjambre de abejas sobre un campo con flores.	47
Figura 3. Símil del el modelado del PSO utilizando como ejemplo cardumen de peces.	48
Figura 4. Componentes de vector velocidad. Movimiento sobre un espacio bidimensional de las partículas i e $i+1$ en la iteración $k+1$	51
Figura 5. Cruce de 1 Punto.	64
Figura 6. Cruce de 2 Punto.	65
Figura 7. Cruce Uniforme.	67
Figura 8. Mutación.	69
Figura 9 Cercha en estudio, planta y perfil.	79
Figura 10. Cercha en estudio, vista 3D.	80
Figura 11 Representación del cromosoma implementado en el programa.	98
Figura 12. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	104
Figura 13. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	105
Figura 14. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	105
Figura 15. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	105
Figura 16. Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	106

Figura 17. Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	106
Figura 18. Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	106
Figura 19. Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	107
Figura 20. Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	107
Figura 21. Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)	107
Figura 22. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	109
Figura 23. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	109
Figura 24. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	109
Figura 25. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	110
Figura 26. Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	110
Figura 27. Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	110
Figura 28. Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	111
Figura 29. Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	111
Figura 30. Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	111
Figura 31. Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)	112

Figura 32. Variación del peso en cada iteración para las tres ejecuciones del algoritmo con un enjambre de 10 partículas (Caso 1)	113
Figura 33. Variación de las áreas en cada iteración para las tres ejecuciones del algoritmo con un enjambre de 10 partículas (Caso 1)	113
Figura 34. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	118
Figura 35. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	118
Figura 36. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	118
Figura 37. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	119
Figura 38. Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	119
Figura 39. Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	119
Figura 40. Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	120
Figura 41. Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	120
Figura 42. Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	120
Figura 43. Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)	121
Figura 44. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	122
Figura 45. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	122
Figura 46. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	123

Figura 47. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	123
Figura 48. Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	123
Figura 49. Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	124
Figura 50. Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	124
Figura 51. Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	124
Figura 52. Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	125
Figura 53. Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)	125
Figura 54. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	126
Figura 55. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	127
Figura 56. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	127
Figura 57. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	127
Figura 58. Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	128
Figura 59. Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	128
Figura 60. Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	128
Figura 61. Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	129

Figura 62. Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	129
Figura 63. Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)	129
Figura 64. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)	131
Figura 65. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)	131
Figura 66. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)	131
Figura 67. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)	132
Figura 68. Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)	132
Figura 69. Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)	133
Figura 70. Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)	133
Figura 71. Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)	133
Figura 72. Variación del peso en cada iteración para la ejecución del algoritmo con un enjambre de 50 partículas (Caso 2).....	134
Figura 73. Variación de las áreas en cada iteración para la ejecución del algoritmo con un enjambre de 50 partículas (Caso 2).....	134
Figura 74. Peso óptimo obtenido para cada población a medida que aumentaban las generaciones Caso 1.	138
Figura 75. Peso óptimo obtenido para cada población Caso1.....	139
Figura 76. Peso óptimo obtenido en cada una de las ejecuciones del algoritmo Caso 1	139
Figura 77. Comparación de la variación del peso con 10 generaciones.....	140

Figura 78. Comparación de la variación del peso con 20 generaciones.....	140
Figura 79. Comparación de la variación del peso con 40 generaciones.....	141
Figura 80. Comparación de la variación del peso con 60 generaciones.....	141
Figura 81. Comparación de la variación del peso con 100 generaciones.....	142
Figura 82. Variación del peso en cada generación para una población de 10 individuos con 10 generaciones.....	143
Figura 83. Variación del peso en cada generación para una población de 10 individuos con 20 generaciones.....	143
Figura 84. Variación del peso en cada generación para una población de 10 individuos con 40 generaciones.....	144
Figura 85. Variación del peso en cada generación para una población de 10 individuos con 60 generaciones.....	144
Figura 86. Variación del peso en cada generación para una población de 10 individuos con 100 generaciones.....	145
Figura 87. Variación del peso en cada generación para una población de 20 individuos con 10 generaciones.....	145
Figura 88. Variación del peso en cada generación para una población de 20 individuos con 20 generaciones.....	146
Figura 89. Variación del peso en cada generación para una población de 20 individuos con 40 generaciones.....	146
Figura 90. Variación del peso en cada generación para una población de 20 individuos con 60 generaciones.....	147
Figura 91. Variación del peso en cada generación para una población de 20 individuos con 100 generaciones.....	147
Figura 92. Variación del peso en cada generación para una población de 40 individuos con 10 generaciones.....	148
Figura 93. Variación del peso en cada generación para una población de 40 individuos con 20 generaciones.....	148
Figura 94. Variación del peso en cada generación para una población de 40 individuos con 40 generaciones.....	149

Figura 95. Variación del peso en cada generación para una población de 40 individuos con 60 generaciones.....	149
Figura 96. Variación del peso en cada generación para una población de 40 individuos con 100 generaciones.....	150
Figura 97. Variación del peso en cada generación para una población de 60 individuos con 10 generaciones.....	150
Figura 98. Variación del peso en cada generación para una población de 60 individuos con 20 generaciones.....	151
Figura 99. Variación del peso en cada generación para una población de 60 individuos con 40 generaciones.....	151
Figura 100. Variación del peso en cada generación para una población de 60 individuos con 60 generaciones.....	152
Figura 101. Variación del peso en cada generación para una población de 60 individuos con 100 generaciones.....	152
Figura 102. Variación del peso en cada generación para una población de 100 individuos con 10 generaciones.....	153
Figura 103. Variación del peso en cada generación para una población de 100 individuos con 20 generaciones.....	153
Figura 104. Variación del peso en cada generación para una población de 100 individuos con 40 generaciones.....	153
Figura 105. Variación del peso en cada generación para una población de 100 individuos con 60 generaciones.....	154
Figura 106. Variación del peso en cada generación para una población de 100 individuos con 100 generaciones.....	154
Figura 107. Peso óptimo obtenido para cada población a medida que aumentaban las generaciones Caso 2.	157
Figura 108. Peso óptimo obtenido para cada población Caso 2.....	157
Figura 109. Peso óptimo obtenido en cada una de las ejecuciones del algoritmo Caso 2.	160

Figura 110. Comparación de la variación del peso con 20 generaciones para el caso 2	160
Figura 111. Comparación de la variación del peso con 40 generaciones para el caso 2	161
Figura 112. Comparación de la variación del peso con 60 generaciones.....	161
Figura 113. Comparación de la variación del peso con 100 generaciones.....	162
Figura 114. Comparación de la variación del peso con 150 generaciones.....	162
Figura 115. Variación del peso en cada generación para una población de 20 individuos con 20 generaciones.....	163
Figura 116. Variación del peso en cada generación para una población de 20 individuos con 40 generaciones.....	164
Figura 117. Variación del peso en cada generación para una población de 20 individuos con 60 generaciones.....	164
Figura 118. Variación del peso en cada generación para una población de 20 individuos con 100 generaciones.....	165
Figura 119. Variación del peso en cada generación para una población de 20 individuos con 150 generaciones.....	165
Figura 120. Variación del peso en cada generación para una población de 50 individuos con 20 generaciones.....	166
Figura 121. Variación del peso en cada generación para una población de 50 individuos con 40 generaciones.....	166
Figura 122. Variación del peso en cada generación para una población de 50 individuos con 60 generaciones.....	166
Figura 123. Variación del peso en cada generación para una población de 50 individuos con 100 generaciones.....	167
Figura 124. Variación del peso en cada generación para una población de 50 individuos con 150 generaciones.....	167
Figura 125. Variación del peso en cada generación para una población de 80 individuos con 20 generaciones.....	168

Figura 126. Variación del peso en cada generación para una población de 80 individuos con 40 generaciones.....	168
Figura 127. Variación del peso en cada generación para una población de 80 individuos con 60 generaciones.....	168
Figura 128. Variación del peso en cada generación para una población de 80 individuos con 100 generaciones.....	169
Figura 129. Variación del peso en cada generación para una población de 80 individuos con 150 generaciones.....	169
Figura 130. Variación del peso en cada generación para una población de 120 individuos con 20 generaciones.....	170
Figura 131. Variación del peso en cada generación para una población de 120 individuos con 40 generaciones.....	170
Figura 132. Variación del peso en cada generación para una población de 120 individuos con 60 generaciones.....	171
Figura 133. Variación del peso en cada generación para una población de 120 individuos con 100 generaciones.....	171
Figura 134. Variación del peso en cada generación para una población de 150 individuos con 20 generaciones.....	172
Figura 135. Variación del peso en cada generación para una población de 150 individuos con 40 generaciones.....	172
Figura 136. Variación del peso en cada generación para una población de 150 individuos con 60 generaciones.....	173
Figura 137. Variación del peso en cada generación para una población de 150 individuos con 100 generaciones.....	173

LISTA DE TABLAS

Tabla 1 Perfiles utilizados en el diseño de la cercha	81
Tabla 2 Coordenadas de los nodos	83
Tabla 3 Enumeración de los miembros.....	84
Tabla 4 Relación entre la cadena binaria y los valores discretos.....	98
Tabla 5 Diseño inicial para agrupación de miembros del Caso 1	101
Tabla 6 Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 1 (Análisis inicial)	103
Tabla 7 Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 1 (Análisis final y peso óptimo)	104
Tabla 8 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 2 partículas (Caso 1).....	108
Tabla 9 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 5 partículas (Caso 1).....	112
Tabla 10 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 10 partículas (Caso 1).....	114
Tabla 11 Comparación del diseño inicial y el diseño óptimo obtenido con el PSO. (Caso 1)	114
Tabla 12 Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 2 (Análisis inicial)	116
Tabla 13 Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 2 (Análisis final y peso óptimo).....	117
Tabla 14 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 2 partículas (Caso 2).....	121
Tabla 15 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 5 partículas (Caso 2).....	126
Tabla 16 Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 10 partículas (Caso 2).....	130

Tabla 17 Diseño óptimo obtenido por el PSO para el Caso 2.....	134
Tabla 18 Porcentaje de reducción del peso entre los dos casos de agrupación de miembros	135
Tabla 19 Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 1(Análisis inicial)	136
Tabla 20 Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 1(Análisis final y peso óptimo).....	137
Tabla 21 Comparación del diseño inicial y el diseño óptimo obtenido con el AG.	155
Tabla 22 Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 2 (Análisis inicial)	158
Tabla 23 Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 2 (Análisis final y peso óptimo).....	159
Tabla 24 Diseño óptimo obtenido por el AG para el Caso 2	174
Tabla 25 Comparación del peso óptimo reportado en el artículo y el resultado óptimo obtenido en este trabajo.....	174
Tabla 26 Especificaciones de los Perfiles utilizados en el diseño.....	175
Tabla 27 Peso y costo para el diseño inicial propuesto por los autores.....	176
Tabla 28 Peso y costo para el diseño optimo para el Caso 1	176
Tabla 29 Peso y costo para el diseño optimo para el Caso 2	177
Tabla 30 Pesos y costos para los tres diseños, y porcentaje de reducción respecto al diseño inicial propuesto.....	177

LISTA DE ANEXOS

ANEXO A	186
----------------------	------------

RESUMEN

TÍTULO:

OPTIMIZACIÓN DEL PESO DE CERCHAS EN 3D MEDIANTE ALGORITMOS GENÉTICOS (GA) Y OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO).*

AUTORES:

JORGE ALBERTO MOLINA MOJICA, EDUARDO ALFONSO PÁEZ ESTEBAN**

PALABRAS CLAVE:

Optimización, Algoritmos Genéticos (AG), Optimización por Enjambre De Partículas (PSO), Cercha 3D.

DESCRIPCIÓN:

Este proyecto consigue implementar metodologías heurísticas como los Algoritmos Genéticos y Optimización por Enjambre de Partículas en el campo de la ingeniería civil, efectuando la optimización estructural.

Para establecer qué se desea obtener en este trabajo, se define optimización como la encargada de proporcionar soluciones con un ahorro de esfuerzo y de tiempo. En este caso se aplican los conceptos de modelación, estructuración y programación de las realidades físicas que se requieran; esta vez se utilizará en el estudio de las cerchas en 3D compuestas exclusivamente de perfiles tubulares metálicos. Es necesario explicar que este modelo no solo se aplica en la cerchas de cubiertas sino que tiene cabida en otros modelos estructurales.

A su vez se implementa el desarrollo de los algoritmos genéticos y de optimización por enjambre de partículas encontrando las soluciones óptimas al desarrollo y diseño de las cerchas proponiendo una mejor solución por agrupación a la propuesta en el artículo *Optimization of 3d trusses with adaptive approach in genetic algorithms*. Vedat Toğan, Ayşe T. Daloğlu. Trabzon, Turkey : *Engineering Structures* 28 (2006) 1019–1027, 18 January 2006, en el cual es estudiado y enfocado el desarrollo de este proyecto para tener una base sólida y sustentada de resultados.

Por último, nos enfocamos en el software desarrollado en el ANEXO A: PROGRAMACIÓN EN MATLAB® (R2009b), que deja la base suficiente para el desarrollo de estructuras metálicas con modificaciones en los programas realizados, base que cumple la norma NSR10 en su título F para el análisis de estructuras metálicas y cumpliendo su análisis dinámico por Newmark y el respectivo análisis estático.

* Proyecto de grado – Modalidad de investigación.

** Facultad de ingenierías Físico-Mecánicas, Escuela Ingeniería Civil, Director: Begambre Carrillo, Oscar.

ABSTRACT

TITLE:

OPTIMIZATION OF THE WEIGHT OF TRUSS 3D BY GENETIC ALGORITHMS (GA) AND PARTICLE SWARM OPTIMIZATION (PSO).*

AUTHOR:

JORGE ALBERTO MOLINA MOJICA, EDUARDO ALFONSO PÁEZ ESTEBAN**

KEY WORDS:

Optimization, Genetics Algorithms (GA), Particle Swarm Optimization (PSO), Truss 3d.

DESCRIPTION:

This project succeeds in implementing heuristic methodologies as the Genetic Algorithms and Optimization for Swarm of Particles in the field of the civil engineering, effecting the structural optimization.

To establish what is desired in this work, optimization is defined as responsible for providing solutions to save effort and time. In this case apply the concepts of modeling, structuring and programming of the physical realities that are required, this time will be used in the study of 3D trusses consist exclusively of tubular metal. It is necessary to explain that this model applies only in the roof trusses but has no place in structural models.

At the same time implements the development of genetic algorithms and particle swarm optimization to find optimal solutions to the development and design of the trusses proposing a better solution to the proposed grouping Article Optimization of 3d trusses with adaptive approach in genetic algorithms. Vedat Togan, Ayşe T. Daloğlu. Trabzon, Turkey, Engineering Structures 28 (2006) 1019-1027, 18 January 2006, which is studied and focused the development of this project to have a solid and sustained results.

Lastly, we focus ourselves in the software developed in the ANNEX TO: PROGRAMMING IN MATLAB® (R2009b) that leaves the enough base for the development of metallic structures with modifications in the carried out programs, base that it completes the norm NSR10 in its title F for the analysis of metallic structures and completing its dynamic analysis for Newmark and the static respective analysis.

Finally, we focus on software developed in APPENDIX A: PROGRAMMING IN MATLAB ® (R2009b), leaving a sufficient basis for the development of metal structures with modifications completed programs that meet the standard based on its title NSR10 F for analysis of metal structures and fulfill its dynamic analysis by Newmark and the respective static analysis.

* Degree Project - Research modality.

** Faculty of Physical-Mechanical Sciences. School of Civil Engineering. Director: Begambre Carrillo, Oscar.

INTRODUCCIÓN

Es la naturaleza del ser humano buscar la mejor solución posible a todo inconveniente que se le presenta en su diario vivir, es evidente que se prefiere la solución cuando además de ser la mejor, es obtenida por un camino más corto; con esto se llega a lo que hoy en día llamamos habitualmente optimización. Vale la pena aclarar que este proyecto no quiere llegar a mecanizar la utilización de un programa, sino, por el contrario aplicar los conceptos que se deben tener claros para el entendimiento y correcta utilización del mismo, que se realizó con el fin de optimizar el peso de cerchas en 3D al aplicar algoritmos genéticos (GA) y optimización por enjambre de partículas (PSO).

Así, como cotidianamente se busca lo mejor en toda ocasión, en este trabajo se aplican los conceptos de optimización a la ingeniería civil y con ello se muestra el interés de obtener el mejor diseño posible de una cercha en 3D. De esta manera, quien lo utilice puede llegar a competir en el mercado de licitaciones con una ventaja sobre los demás diseños porque al requerir menos material estructural se reducen los costos.

A medida que pasa el tiempo es mayor la dificultad de la optimización de los problemas porque estos presentan mayor complejidad o ya han tenido mejoras día tras día, es por ello que aparecen metodologías nuevas y cada vez más eficientes para su optimización. En este sentido se emplean en el presente estudio los métodos de optimización por los algoritmos computacionales AG y PSO, demostrando así, su eficiencia y comprobando cual presenta mejores resultados para este proyecto.

Para establecer que se desea obtener en este trabajo, hay que definir qué es optimización y en qué ámbito se requiere aplicar. Para ello, cabe aclarar, que la optimización se encarga de proporcionar soluciones con un ahorro de esfuerzo y de tiempo. En este caso se aplican los conceptos de modelación, estructuración y programación de las realidades físicas que se requieran, esta vez se utilizará en el estudio de las cerchas en 3D compuestas exclusivamente de perfiles tubulares metálicos. Cabe aclarar que este modelo no sólo se aplica en la cerchas de cubiertas sino que tiene cabida en otros modelos estructurales.

Este proyecto se enfoca en dos métodos heurísticos de optimización como son la optimización por enjambre de partículas (PSO) y los algoritmos genéticos (GA), a los cuales se les hace un preámbulo breve a continuación. El método de optimización de enjambres de partículas (PSO) consiste en un algoritmo iterativo basado en una población de individuos denominada enjambre, en la que cada individuo conocido por partícula, se dice que sobrevuela el espacio de decisión en busca de soluciones óptimas. Por otra parte los algoritmos genéticos generan una relación entre el conjunto de soluciones de un problema y los individuos que componen el mismo, codificando la información de cada solución en una cadena, generalmente binaria, y así estudiar las características al sufrir procesos de mutación y sobrecruzamiento de estos individuos, que para la aplicación a este proyecto serán las componentes de las cerchas, siempre dependiendo del principio de la sobrevivencia del más fuerte, puesto que estos serían los componentes indispensables de la estructura.

Una vez programado los algoritmos de optimización se crea una base de datos para posteriores diseños, esto implica una reducción de esfuerzos en el personal encargado en el diseño de la estructura. Además, el costo de los materiales – que

es uno de los principales factores económicos en la construcción- se reduce a la hora de disminuir el peso o el volumen del sistema estructural.

Este trabajo muestra, inicialmente, los estudios previos realizados a las metodologías que se emplearán más adelante. Esto es lo que conforma el primer capítulo. En el capítulo dos se llega a la explicación clara de lo que se quiere con este trabajo, emulando los conceptos de optimización y sus aplicaciones, tanto las metodologías clásicas como las heurísticas que se desarrollan en este ámbito, y llegar hasta el punto de enfatizar en los métodos de optimización por enjambre de partículas y algoritmos genéticos que son la base del desarrollo de este proyecto.

El tercer capítulo muestra la investigación que se realizó respecto a la optimización de las cerchas en 3D, exponiendo el estudio del diseño que se efectuó por la nueva normativa a implementar en el país y con esto los respectivos modelos matemáticos manejados. De igual forma se muestra la estructura y optimización en la que se enfoca este trabajo que es tomado del artículo *Optimization of 3d trusses with adaptive approach in genetic algorithms. Vedat Toğan, Ayşe T. Daloğlu. Trabzon, Turkey : Engineering Structures 28 (2006) 1019–1027, 18 January 2006*, describiendo los algoritmos implementados para los dos métodos y casos desarrollados. En el cuarto capítulo se muestran los resultados obtenidos y se hace la comparación entre los diferentes casos en estudio.

Finalmente, el quinto capítulo está dedicado a las conclusiones y recomendaciones especificando el aporte propuesto para la disminución del costo en el presupuesto de la estructura estudiada.

OBJETIVOS

OBJETIVO GENERAL

Evaluar el desempeño de los algoritmos GA y PSO en el problema de optimización de una cercha 3D bajo las restricciones de la norma NSR 10, Título F.

OBJETIVOS ESPECÍFICOS

- Programar los métodos GA y PSO en lenguaje Matlab®.
- Determinar la precisión y estabilidad de los métodos utilizados para la optimización de la cercha, GA y PSO.
- Comparar los resultados obtenidos mediante (GA y PSO) con otros métodos reportados en la literatura.
- Comparar los costos estimados del material de la alternativa óptima y la alternativa inicial.

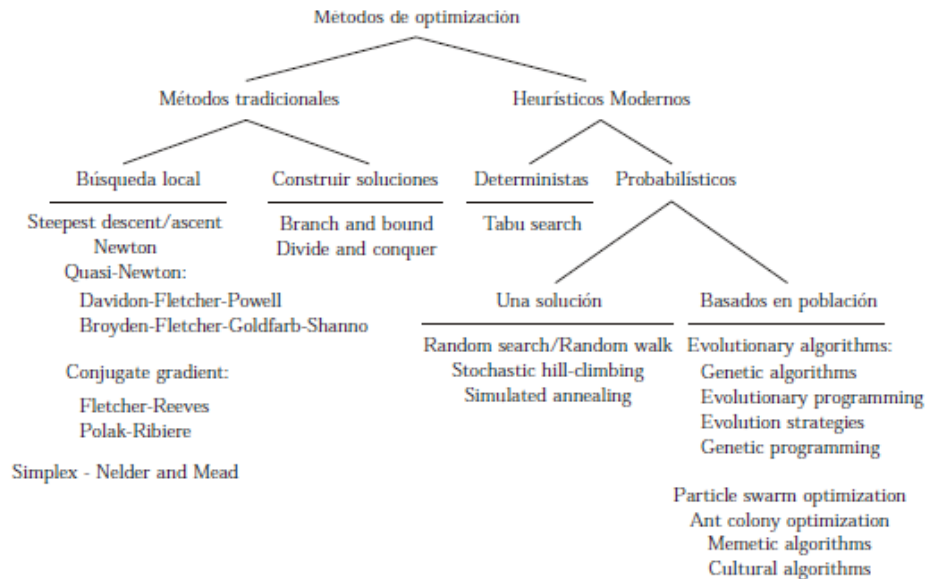
1. GENERALIDADES Y ENFOQUE PRECEDENTE

De forma genérica, puede definirse la optimización como la encargada de determinar las mejores soluciones a problemas matemáticos, que a menudo modelan una realidad física. Los problemas complejos de optimización multidimensional, no lineal y altamente multimodal, pueden encontrarse en ingeniería, economía, geofísica y prácticamente en todos los campos de la ciencia. En este aspecto, la optimización numérica ha adquirida mucha atención entre la comunidad científica, durante las ultimas décadas. Teniendo en cuenta que un aspecto esencial de la optimización heurística es poder decidir que algoritmo y que parámetros de ese algoritmo se ajustan mejor a las características del problema físico bajo análisis, este trabajo pretende, entre otras cosas, dar a los ingenieros y diseñadores herramientas para afrontar esta tarea.

Se han empleados estas metodologías de optimización para el estudio de problemáticas en diferentes ámbitos de la ciencia, entre estos se pueden nombrar la optimización del problema del cajero viajante, en el cual se han enfatizado inicialmente estos estudios, de igual forma se puede hablar de la optimización para estructuras de hormigón y metálicas. Esta optimización en dichas estructuras se logra aplicando los algoritmos a la ingeniería civil. A la vez se puede hacer mención de los trabajos en los cuales se estudiaron las metodologías y sus desarrollos.

Es necesario resaltar el aumento en la aparición de nuevas técnicas y metodologías para afrontar los procesos de diseño, análisis e interpretación de cálculos en el campo, por ello se muestra en la figura 1 los métodos con mas relevancia y bases para futuras investigaciones.

Figura . Clasificación de los métodos de optimización más relevantes. Por uniformidad con la literatura científica se conserva la denominación original en inglés de cada método.



Fuente: Pérez López, J. R. (Octubre de 2005). Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo.

La búsqueda Tabú (Tabu search), de naturaleza determinista, tiene capacidad para escapar de los mínimos o máximos locales aprovechando el conocimiento acerca del dominio de búsqueda y actualizando la solución en curso con el mejor punto de su vecindad. Por otra parte, dentro de los métodos heurísticos de naturaleza probabilística, denominación asociada con el hecho de que la optimización depende de eventos aleatorios, existen dos familias diferenciadas: aquellos que utilizan un único punto de partida y aquellos que utilizan una población.

Los primeros estudios dentro del área de los AG, fueron desarrollados por Joan Holland en 1975. La principal ventaja de los AG radica en que son capaces de evadir los extremos locales de las funciones de costo, pero tienen en su contra la

cantidad de operaciones que hay que realizar. Existen varios puntos en los que un AG difiere de los métodos tradicionales de búsqueda y optimización. Un algoritmo genético trabaja sobre parámetros codificados llamados individuos de la variable que se quiere optimizar, que inicialmente son generados aleatoriamente. Busca la solución mediante una aproximación de una población de puntos, en lugar de una aproximación de punto por punto como hacen los otros métodos. Utiliza reglas probabilísticas de búsqueda en lugar de reglas determinísticas. Trabaja directamente sobre la función y no sobre la derivada ni ninguna otra función auxiliar. Tiene tres operadores básicos que son: selección, cruce y mutación. (Cabezas Rivas, 2002)

En contraste con los algoritmos genéticos, lo propuesto por Kennedy en sus estudios (J. Kennedy, Particle swarm optimization, December 1995) la optimización con enjambre de partículas, la población tiene memoria, es decir, la optimización se dirige y encauza influida por la historia pasada, por la memoria de cada individuo y por el estado presente en el que cada uno se encuentra. Este método ha sido aplicado al diseño de antenas. Otras áreas incluyen, por ejemplo, la aplicación del algoritmo al entrenamiento de redes neuronales o a la planificación de red en servicio de telecomunicación. (Pérez López, Octubre de 2005)

Con diferentes aspectos, pero bajo unos principios similares, Dorigo y otros proponen la así denominada optimización con una colonia de hormigas o ant colony optimization (Dorigo & Di Caro, 1999). Básicamente, los principios del método se limitan a imitar el desplazamiento de las hormigas sobre lo que ahora es un espacio de soluciones, teniendo en cuenta que en su desplazamiento las hormigas trazan unos caminos de feromona que se disipa con el tiempo y la distancia.

2. LA OPTIMIZACIÓN

2.1 GENERALIDADES

Recurrentemente resolvemos pequeños problemas que sólo necesitan métodos sencillos para solucionarlos más que los propuestos por nuestro cerebro para encontrar un buen camino a la solución, pero cada vez aumenta la dificultad de los problemas al aparecer y con esto la complejidad de los sistemas para solucionarlos, es aquí donde surge la optimización. La optimización nace de la necesidad del ser humano de hacer las cosas lo mejor posible, surge del deseo de competitividad de cada ser, ya que el valor que se le da al mejor resultado, mejor interpretación o mejor solución repercute en la búsqueda de mejorar lo anteriormente hecho o propuesto, es aquí donde toma gran significado y valor, el optimizar las situaciones particulares de nuestro entorno.

En la optimización se da un conjunto de problemas que son fundamentales para las disciplinas de ciencias de la computación y la ingeniería. La investigación en esta área tiene por objeto el desarrollo de técnicas eficientes para encontrar los valores mínimos y máximos de una función de variables independientes, es por ello que dada la evolución de los seres podemos encontrar cada vez más soluciones y opciones de aplicación para desarrollar nuestras actividades y cálculos del diario vivir en los campos que requerimos; que en este proyecto se enfocan hacia la ingeniería civil, y los métodos a utilizar son los que ayudan a encontrar y minimizar costos en la construcción, aclarando que en el presente proyecto se guía a las construcciones metálicas enfatizando en las cerchas que utilizan perfiles tubulares cumpliendo la normativa colombiana y las condiciones de diseño adecuadas.

En las secciones siguientes se abordan conceptos básicos y precursores para la optimización.

2.2 OPTIMIZACIÓN CLÁSICA

Desde la década de los sesenta, la programación lineal (PL) ha sido aplicada en diversas áreas de la vida como por ejemplo: sistemas militares, agrícolas, económicos, de transporte y de salud. La PL ofrece bases importantes en el desarrollo de métodos de solución de otras técnicas de la Investigación de operaciones, como lo son la programación entera, la estocástica y la no lineal (Cruz Chávez, Agosto 18 del 2000).

La PL juega un papel muy importante en el estudio de los problemas continuos de optimización considerados como la frontera de los problemas de optimización combinatoria, ya que en los continuos se tienen las características necesarias para que sean considerados dentro del tipo combinatorio. Un problema de optimización combinatoria siempre se le involucra un conjunto de instancias, donde cada una de ellas cuenta con un conjunto finito de posibles soluciones (característica imprescindible de los problemas continuos). Por otra parte, la teoría de optimización clásica se usa para la obtención de los máximos y mínimos de funciones no lineales restringidas y no restringidas, en los que se hace uso del cálculo diferencial.

La teoría de la optimización clásica usa el cálculo diferencial en la optimización de los puntos máximos y mínimos para funciones con y sin restricciones. El método tal vez no sea adecuado para cálculos eficientes o procesos en los cuales se tienen metodologías aportadas de las investigaciones desarrolladas en cada uno

de los campos al aplicar la optimización, para resaltar la optimización clásica cabe resaltar que la teoría que lo fundamenta proporciona las bases para desarrollar la mayor parte de los algoritmos, de programación lineal.

Proporcionando los datos necesarios para la aplicación de metodologías clásicas de optimización se puede recalcar en que si la restricción no existe, o es una restricción de igualdad, con menor o igual número de variables que la función objetivo entonces, el cálculo diferencial, da la respuesta, ya que sólo se trata de buscar los valores extremos de una función.

2.2.1 Método de Newton

El encontrar la solución a un sistema de ecuaciones no lineal es mucho más difícil que el de un sistema lineal. El método de Newton es un procedimiento iterativo y permite la linealización de un sistema de ecuaciones no lineal, para posteriormente, darle solución por cualquier método numérico de ecuaciones lineales simultáneas, este método forma parte de aquellos conocidos como métodos de gradiente. Un sistema de n ecuaciones con n incógnitas (x_1, x_2, \dots, x_n) , se conoce como no lineal, si una o más de estas no es lineal.

De manera general, la solución de un sistema de n ecuaciones no lineales aplicando el método de Newton (Conte & Boor, 1987) se plantea como sigue en la ecuación 1:

Donde f representa la función objetivo y ΔZ_i las variaciones en su respectivo vector.

$$\begin{bmatrix} \frac{\partial f_1(x,y,z,\dots)}{\partial x} & \frac{\partial f_1(x,y,z,\dots)}{\partial y} & \frac{\partial f_1(x,y,z,\dots)}{\partial z} & \dots \\ \frac{\partial f_2(x,y,z,\dots)}{\partial x} & \frac{\partial f_2(x,y,z,\dots)}{\partial y} & \frac{\partial f_2(x,y,z,\dots)}{\partial z} & \dots \\ \frac{\partial f_3(x,y,z,\dots)}{\partial x} & \frac{\partial f_3(x,y,z,\dots)}{\partial y} & \frac{\partial f_3(x,y,z,\dots)}{\partial z} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(x,y,z,\dots)}{\partial x} & \frac{\partial f_n(x,y,z,\dots)}{\partial y} & \frac{\partial f_n(x,y,z,\dots)}{\partial z} & \dots \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \\ \vdots \end{bmatrix} \quad ()$$

Evaluado en (X_i, Y_i, Z_i, \dots) y resolviendo se tiene:

$$X_{i+1} = X_i + \Delta X_i, Y_{i+1} = Y_i + \Delta Y_i, Z_{i+1} = Z_i + \Delta Z_i, \dots$$

Sea la expresión ecuación 2 la suma de todas las ecuaciones no lineales que representan el sistema en estudio

$$f_{2n}(\underline{X}^{k+1}) = D_n(\underline{X}^{k+1}) + E_n(\underline{X}^{k+1}) \quad ()$$

La iteración de Newton se podrá obtener para el conjunto de ecuaciones a partir de la ecuación 2, usando la serie de Taylor en una aproximación de primer orden

$$f_{2n}^{k+1}(\underline{X}_{i+1}) \approx f_{2n}^{k+1}(\underline{X}_i) + f'_{2n}{}^{k+1}(\underline{X}_i)(\underline{X}_{i+1} - \underline{X}_i) = 0 \quad ()$$

La ecuación 3 representa un conjunto de $2n$ ecuaciones lineales acopladas para el vector de incógnitas X en el paso de tiempo $k+1$, en la iteración $i+1$. Los términos en derivadas darán lugar a la formación de la matriz Jacobiana:

$$f'_{2n}{}^{k+1}(\underline{X}_i) = \sum \frac{\partial f_{2n}{}^{k+1}(\underline{X}_i)}{\partial \underline{X}_i} \quad ()$$

La solución del conjunto de ecuaciones lineales encontradas en cada iteración (matriz Jacobiana) será resuelta por cualquier método lineal de matrices hasta que los residuales sean menores a una tolerancia designada, muy próxima a cero, ecuación 3. (Cruz Chávez, Agosto 18 del 2000)

2.2.2 El método simplex¹

Para resolver un modelo de PL por medio del método simplex, este se deberá poner en forma estándar aplicando los siguientes puntos:

1. Todas las restricciones son ecuaciones con un segundo miembro positivo.
2. Una restricción se convierte en ecuación adicionando una variable de holgura (o restando una variable de exceso) al primer miembro de la ecuación. El segundo miembro de la ecuación se puede hacer positivo, multiplicando ambos lados por menos uno (-1). Y la dirección de una desigualdad se invierte multiplicando ambos miembros de la ecuación por -1.
3. Todas las variables son positivas.
4. En el caso de que una variable y_i cualquiera, sea irrestricta (no restringida), esta pudiera expresarse en términos de dos variables positivas mediante el uso de la sustitución. $y_i = y'_i - y''_i$ donde $y'_i, y''_i \geq 0$.

¹ Para mayor información sobre este método se puede leer Cruz Chávez, M. A. (Agosto 18 del 2000). *Máximos y mínimos*. Cuernavaca Morelos, México: www.uaem.mx/posgrado/mcruz/maximominimo.PDF; UAEM Av. Universidad Chamilpa C.P.

5. La función objetivo puede ser de maximización o de minimización.

2.2.3 Otros métodos

Los métodos abordados en las secciones anteriores para optimización clásica son los precursores y base de las metodologías actualmente desarrolladas y que conllevan un toque de innovación. Entre otros métodos clásicos vale la pena resaltar el Lagrangiano -método de derivadas restringidas-; el Jacobiano -métodos de estudios numéricos básicos- y en programación lineal, podemos referenciar el método de ascenso pronunciado (steepest ascent) (Cruz Chávez, Agosto 18 del 2000) en donde la terminación del método del gradiente se realiza en el punto donde el gradiente se vuelve nulo. Esta es solamente una condición necesaria para la optimalidad. Así, se enfatiza que la optimalidad no puede verificarse al menos que se conozca a priori que $f(\mathbf{X})$ es cóncava o convexa. Finalmente, en los métodos clásicos de optimización se acotan los métodos numéricos existentes y aquellos que se componen de teoremas básicos como la prueba y el error tanto como los seguidos por las funciones iterativas.

2.3 OPTIMIZACIÓN HEURÍSTICA

Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como *el arte y la ciencia del descubrimiento y de la invención* o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente.

El matemático George Pólya en 1945, con su libro *Cómo resolverlo (How to solve it)* fue quien difundió el término y su concepto, habiendo estudiado tantas pruebas

matemáticas desde su juventud, quería saber cómo los matemáticos llegan a ellas, pues es desde allí donde se llega a la implementación de nuevos conceptos, es por esto que, su libro alberga los pasos heurísticos que trataba de enseñar a sus alumnos de matemáticas. Cuatro patrones expuestos en él, ilustran el concepto mejor que ninguna definición:

- Si no consigue entender un problema, dibuja un esquema.
- Si no encuentra la solución, haga como si ya la tuviera y mire qué puede deducir de ella (*razonando a la inversa*).
- Si el problema es abstracto, pruebe examinar con un ejemplo concreto.
- Intente abordar primero un problema más general (es la “paradoja del inventor”: el propósito más ambicioso es el que tiene más posibilidades de éxito).

En este contexto, una metaheurística es un método para resolver un tipo de problema computacional general, usando los parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera eficiente. La formación de esta palabra de sus significativos griegos haciendo referencia a encontrar respuestas en niveles superiores, dando a entender que son métodos para encontrar las mejores soluciones a problemáticas de ingeniería avanzada y de necesidades lógicas como el buscar siempre ahorrar recursos y esfuerzos.

En ingeniería, los métodos heurísticos están basados en la experiencia que puede utilizarse como ayuda para resolver problemas de diseño, desde calcular los recursos necesarios hasta en planear las condiciones de operación de los sistemas. Mediante el uso de heurísticas, es posible resolver más rápido problemas conocidos o similares a otros conocidos. Existen varios métodos

heurísticos disponibles para los ingenieros como, por ejemplo, el análisis modal de fallos y efectos depende de un grupo de ingenieros experimentados que evalúan los problemas y fallos, los ordenan según su importancia y recomiendan soluciones.

2.3.1 Algoritmos de optimización

Fue John Holland en los años setenta el primero que pretendió llevar esta estrategia de actuación al campo de la computación y tratar de diseñar sistemas artificiales con un comportamiento análogo al del sistema evolutivo natural. Este intento trajo consigo el nacimiento de lo que hoy conocemos como algoritmos evolutivos. Un algoritmo evolutivo es un algoritmo probabilístico que mantiene una población de individuos, $P(t)$ para cada iteración t . Cada uno de los individuos constituye una potencial solución del problema representada mediante alguna estructura de datos. Cada una de estas potenciales soluciones es evaluada para obtener una medida de su calidad como solución. Posteriormente, se construye una nueva población (iteración $t+1$) mediante un proceso de selección en $P(t)$ de manera que los mejores individuos tienen mayor probabilidad de ser seleccionados. A continuación se altera esta nueva población mediante la aplicación sobre algunos miembros de la misma de ciertos operadores genéticos. Estos operadores genéticos pueden ser unarios (mutaciones), produciendo nuevos individuos mediante pequeños cambios en el individuo sobre el que se aplican, o cruces de manera que dos o más individuos se combinan para producir nuevos individuos. Así se obtiene finalmente una nueva población $P(t+1)$. Tras un número de iteraciones o generaciones es de esperar que el mejor individuo de la población final obtenida represente una solución del problema óptima o al menos razonablemente buena.

2.3.2 Optimización por enjambre de partículas (PSO)

Encontrada en la literatura científica como Particle Swarm Optimization (PSO) ó comúnmente optimización por enjambre de partículas, parte al igual que las demás técnicas estocásticas de cálculo evolutivo, con procedimientos que son bioinspirados, imitando y mimetizando el comportamiento de procesos naturales, quieren aprovechar estos análisis ventajosos para concurrencia de partículas. Actualmente en el área de inteligencia computacional se cuenta con dos métodos que han tomado gran importancia que son el PSO y la optimización con una colonia de hormigas o en literatura científica Ant Colony Optimization (ACO), ambos mimetizando comportamientos de interacción de los individuos en sus entornos y con sus similares.

2.3.2.1 Generalidades

Los estudios que comenzaron Kennedy y Eberhart, (J. Kennedy, Particle swarm optimization, December 1995) , se basan en la representación de ciertos grupos de individuos en especial, así tales como las bandadas de pajaros, cardúmenes de peces, enjambres de abejas y colonias de hormigas, primordialmente sin descartar cualquier otro grupo de individuos que presentan comportamientos sociales entre sí.

El movimiento de cada una de las partículas se sintetiza el recuerdo que ellas presentan de la iteración inmediatamente anterior y con ello fijan un nuevo punto de seguimiento mejor o igual que el anterior pero nunca haciendo un retroceso en

la meta que se desea, esta memoria que presenta cada individuo se puede decir que es autobiográfica y va de la mano con el impulso que se tiene en el enjambre por seguir a quien tiene un mejor camino o mejor solución.

Los orígenes del PSO como método estocástico de optimización global se remontan a los estudios iniciados por Kennedy y Eberhart en (J. Kennedy, Particle swarm optimization, December 1995), quienes se fijan como objetivo inicial simular gráficamente el movimiento sincronizado e impredecible de grupos tales como los bancos de peces o las bandadas de aves, intrigados por la capacidad de estos grupos para separarse, reagruparse o encontrar alimento. En línea con trabajos previos en el ámbito de la biología y de la sociología, que concluyen que el comportamiento, inteligencia y movimiento de estas agrupaciones, entre las cuales se podría incluir con un cierto grado de abstracción a los seres humanos, está relacionado directamente con la capacidad de los individuos para compartir información y aprovecharse de la experiencia acumulada por sus congéneres, en (J. Kennedy, Particle swarm optimization, December 1995) se modela dicho comportamiento de forma matemática utilizando expresiones simples que revelan su potencial como método de optimización.

En la terminología utilizada en PSO, Kennedy y Eberhart, (J. Kennedy, Particle swarm optimization, December 1995), (J. Kennedy, Swarm intelligence, 2001), introducen el término general partícula o agente para representar a los peces, pájaros, abejas, hormigas o cualquier otro tipo de individuos que exhiban un comportamiento social como grupo, en forma de una colección de agentes que interactúan entre sí. De acuerdo con los fundamentos teóricos del método, el movimiento de cada una de estas partículas hacia un objetivo común en dos dimensiones está condicionado por dos factores básicos, la memoria autobiográfica de la partícula o nostalgia y la influencia social de todo el enjambre. A nivel computacional, como método de optimización, esta filosofía puede

extenderse a un espacio N-dimensional de acuerdo con el problema bajo análisis. La posición instantánea de cada una de las partículas de la población en el espacio N-dimensional representa una solución potencial, siendo N el número de incógnitas del problema original. Básicamente, el proceso evolutivo se reduce a mover cada partícula dentro del espacio de soluciones con una velocidad que variará de acuerdo a su velocidad actual, a la memoria de la partícula y a la información global que comparte el resto del enjambre, utilizando una función de fitness para cuantificar la calidad de cada partícula en función de la posición que ésta ocupe.

Más allá de la propia naturaleza del método, los esquemas existentes para su implementación son muy diversos. Dependiendo de cómo se actualicen las posiciones de las partículas surgen las versiones síncrona y asíncrona del algoritmo. Adicionalmente, dependiendo de cómo se haga influir la experiencia acumulada por el enjambre sobre el movimiento de cada una de las partículas que lo integran, se puede distinguir entre PSO local y global. La combinación de estas cuatro variantes resume los esquemas desarrollados e investigados en este capítulo, siendo a su vez los más comúnmente utilizados. No obstante, en la literatura están muy extendidas múltiples variantes alternativas a los esquemas convencionales, en la mayoría de los casos fruto de modificaciones introducidas por los propios autores para mejorar el rendimiento del algoritmo original en aplicaciones concretas (Pérez López, Octubre de 2005).

Trasladando la filosofía de PSO al campo de la vida artificial y del cómputo evolutivo, este método guarda cierta analogía con los GA. Sin embargo, su extremada simplicidad de programación y el menor número de parámetros de los que depende el funcionamiento interno del algoritmo, le convierten en una alternativa atractiva a los GA, de reciente aplicación y difusión en diferentes campos aplicables a las ingenierías como el electromagnetismo.

Se realizará un análisis en profundidad de los principios del PSO como método de optimización. Se comienza presentando los fundamentos de PSO, para realizar a continuación un análisis de los esquemas que se han implementado y un análisis paramétrico con un doble objetivo: por un lado poner a punto el código fuente desarrollado y, por otro, analizar la influencia de los principales parámetros del algoritmo sobre la convergencia de la optimización, buscando un compromiso entre precisión y coste computacional. Se enfatizará el método como herramienta de solución de los problemas dados en la ingeniería.

2.3.2.2 Fundamentos del movimiento de partículas

Para aplicaciones en el ámbito de la vida artificial se deben respetar cinco principios básicos de lo que se entiende como inteligencia de grupo, más comúnmente recogido en la literatura como *swarm intelligence* (J. Kennedy, Particle swarm optimization, December 1995), (Millonas, 1994). Estos principios reciben el nombre de proximidad, calidad, diversidad de respuesta, estabilidad y adaptabilidad. En el presente apartado se justifica el modo en el que el PSO se adhiere a estos cinco postulados (Pérez López, Octubre de 2005).

De acuerdo con el primer principio, *proximidad*, la población debiera ser capaz de realizar cálculos sencillos de espacio y tiempo, lo cual se traduce en PSO a movimientos en N dimensiones llevadas a cabo durante una serie de intervalos de tiempo que coinciden con movimientos de la población a una determinada velocidad (Pérez López, Octubre de 2005).

El segundo principio, *calidad*, promueve la capacidad de la población para responder a factores de calidad en el espacio de soluciones, lo que se consigue

en PSO con la memoria de la partícula y con la historia o conocimiento social que comparten entre sí todos los congéneres (Pérez López, Octubre de 2005).

El tercer principio *diversidad de respuesta*, promueve esta misma dentro de la población, y está garantizado en PSO por las diferentes tendencias marcadas por la memoria personal de cada partícula y por la historia de la mejor posición visitada por todo el conjunto (Pérez López, Octubre de 2005).

El cuarto y quinto principio resaltan aspectos contrapuestos. Por un lado, la población sólo cambia su comportamiento como grupo cuando se actualiza la mejor posición históricamente visitada por alguno de los miembros que lo integran, respetando así el principio de *estabilidad*. Por el contrario, de acuerdo con el quinto principio, la población debe de ser a su vez *adaptativa*, es decir, debe ser capaz de modificar su comportamiento y movimiento cuando hay alguna señal que así lo recomienda desde el punto de vista de ahorro computacional o de mejora en la precisión. Esta premisa se consigue fácilmente en PSO, dado que la población en su conjunto cambia su rumbo cuando alguna de las partículas alcanza una solución global que mejora el resultado (Pérez López, Octubre de 2005) y economiza esfuerzos.

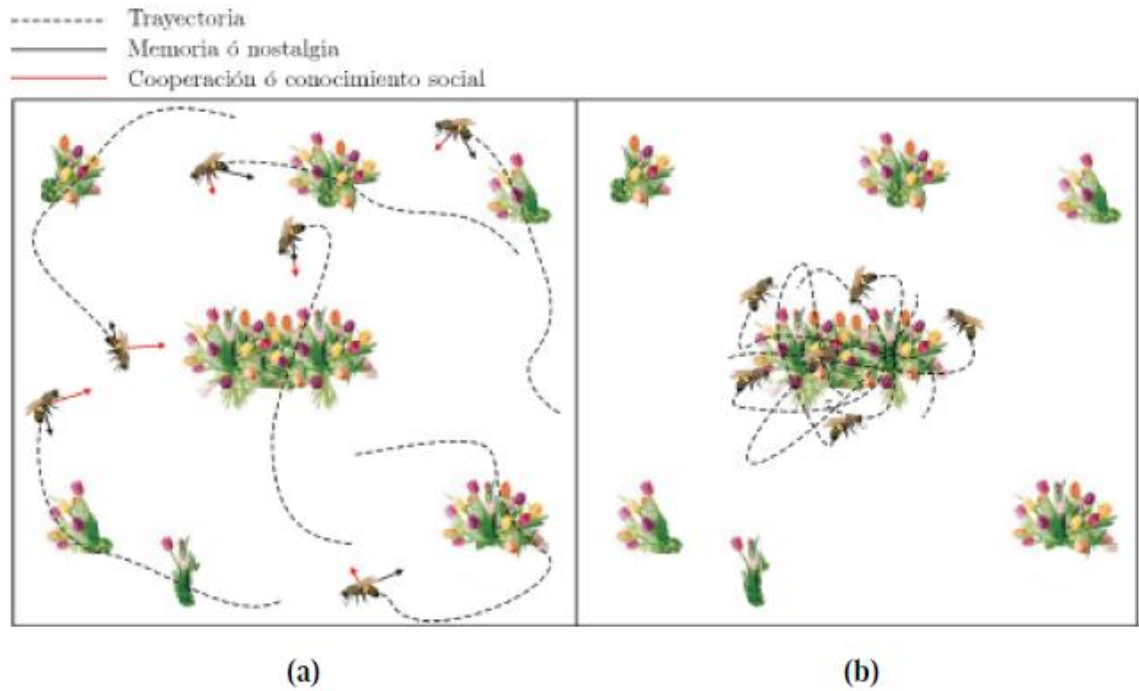
2.3.2.3 El comportamiento social como método de optimización global

En PSO se usa el término multitud, enjambre y de forma genérica swarm, para referirse a cualquier conjunto de agentes o individuos que interactúan entre sí y con el medio que les rodea. El ejemplo clásico de enjambre lo representan las

abejas en las inmediaciones de la colmena así como los cardúmenes de peces, aunque esta analogía se hace extensible a cualesquiera otros sistemas con una arquitectura y comportamiento social como grupo similar (Pérez López, Octubre de 2005).

Para entender el funcionamiento del PSO como algoritmo de optimización supongamos, a modo de ejemplo, el comportamiento que exhibe un enjambre de abejas en su movimiento sobre un campo cubierto con diferentes concentraciones de flores. Sin ningún conocimiento a priori del espacio de búsqueda, las abejas inician su movimiento desde posiciones aleatorias y con velocidades aleatorias. En su desplazamiento, el objetivo del enjambre se centra en encontrar el emplazamiento con la mayor densidad de flores. Cada abeja tiene memoria y puede recordar la posición visitada con mayor densidad de flores y también conoce, por mecanismos de comunicación con sus congéneres, la localización donde otras abejas encontraron una densidad de flores significativa. Esta dupla de información es utilizada por la abeja para modificar continuamente su trayectoria, acelerando en ambas direcciones y volando hacia un punto espacial intermedio que dependerá de su posición actual y de cómo influyan sobre su decisión las así denominadas nostalgia o memoria y cooperación o conocimiento social. De esta forma, las abejas se encuentran permanentemente sobrevolando el campo en busca de posiciones con mayor densidad de flores, redirigiendo en parte la trayectoria del enjambre cada vez que se encuentran configuraciones de mayor calidad. Con el transcurso del tiempo, una vez ha sido explorado el espacio de soluciones en su totalidad, el conjunto del enjambre se encontrará volando alrededor de la zona con la mayor concentración de flores de todo el campo. En esta situación, las abejas, incapaces de encontrar posiciones alternativas mejores, son permanentemente atraídas hacia dicha posición (Pérez López, Octubre de 2005). Este comportamiento social se muestra en las Figuras 2 y 3.

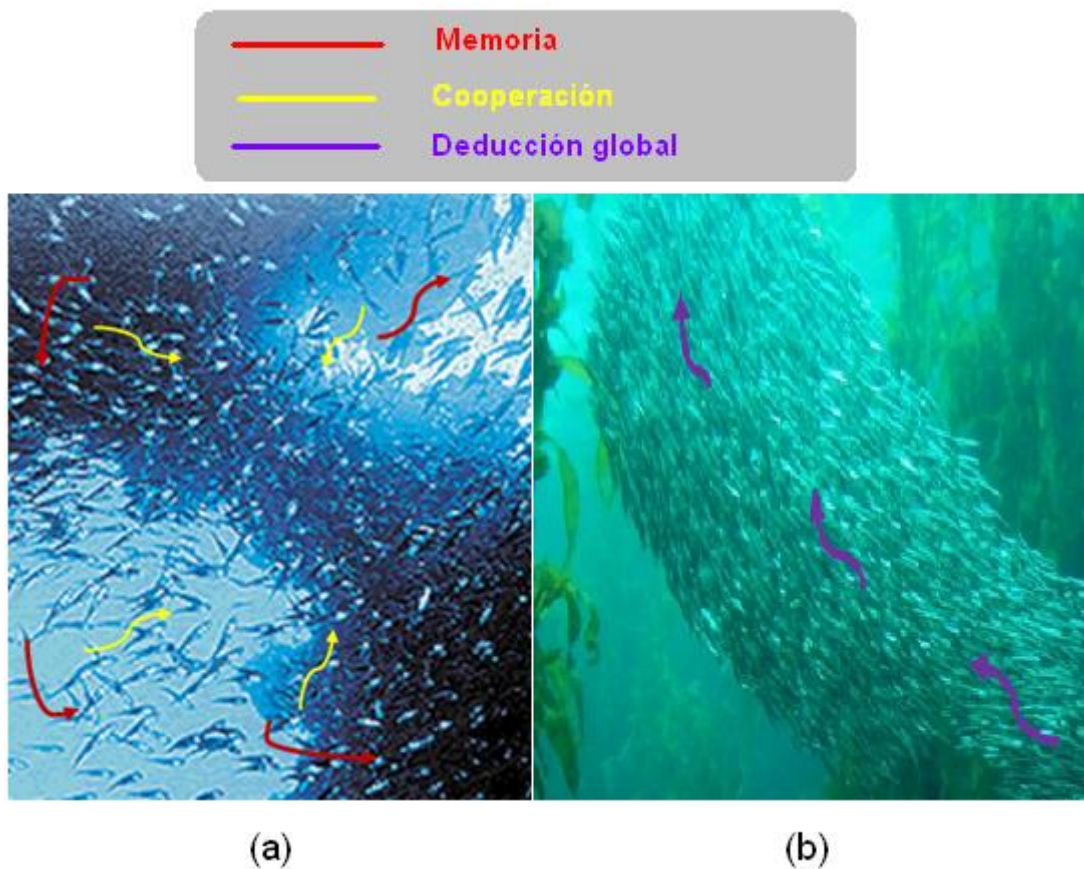
Figura . Modelado del PSO utilizando como símil el movimiento de un enjambre de abejas sobre un campo con flores.



Fuente: Pérez López, J. R. (Octubre de 2005). Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo.

Donde, (a) en su desplazamiento, las abejas son atraídas hacia las zonas de mayor concentración de flores encontradas personalmente por cada individuo (memoria) y por el conjunto del enjambre (cooperación). (b) Una vez que las abejas han sido atraídas a la zona con mayor concentración de flores, que equivale en términos de PSO a la convergencia hacia una solución global, éstas permanecen sobrevolando dicha zona con velocidades muy reducidas (Pérez López, Octubre de 2005).

Figura . Símil del el modelado del PSO utilizando como ejemplo cardumen de peces.



Fuente: Elaboración propia

Donde, (a) En su movimiento, los peces se orientan hacia la zona de mayor concentración de comida encontrada por cada individuo y por el cardumen de peces, memorizando su recorrido para tener guía si a medida que avanzan hacia donde se encuentra la mayoría del cardumen de peces obtienen mejores resultados. (b) Ya encaminados a la zona con mayor concentración del cardumen de peces, encuentran la mayor concentración de comida, con esto haciendo una deducción global que en PSO se refleja al encaminarse a la solución óptima, y para mantenerse en este espacio, lo rodean con velocidades muy reducidas para no apartarse del mismo.

El comportamiento social que exhiben estos y otros organismos se puede entender como un método de optimización en el que el espacio de búsqueda se puede extender a las N dimensiones del problema a optimizar, y donde cada partícula, abeja en este caso, se identifique como una posible solución potencial al problema, caracterizada por un vector velocidad y un vector posición, ambos en N dimensiones. El problema se reduce a establecer la ecuación que dicte como debe moverse cada partícula de la población en el espacio N-dimensional para mimetizar la inteligencia de estas comunidades y evitar a su vez caer en soluciones locales (Pérez López, Octubre de 2005).

2.3.2.4 Operador velocidad y parámetros del algoritmo

En la formulación de PSO se define la velocidad de partícula como el único operador disponible para controlar la evolución de la optimización. De forma genérica, supongamos una población de I partículas donde cada partícula del enjambre se identifica con dos variables de estado inicializadas aleatoriamente dentro del espacio N-dimensional que establece el problema a optimizar: un vector velocidad, $V_1 = (v_{i1}, v_{i2}, \dots, v_{iN})$, y un vector de posición, $X_1 = (x_{i1}, x_{i2}, \dots, x_{iN})$. Este último corresponde a una solución potencial al problema de optimización. Los límites de los parámetros a optimizar, $x_n \in [x_{n,min}, x_{n,max}]$ con $x_n \in \cdot$, conforman en su conjunto el espacio de búsqueda al cual debe restringirse el movimiento del enjambre.

Adicionalmente, cada partícula mantiene en memoria información de la posición espacial asociada con la mejor solución históricamente visitada por ésta, $P_1 = (p_{i1}, p_{i2}, \dots, p_{iN})$, y también conoce la posición de la mejor partícula o solución encontrada por todos sus congéneres, $G = (g_1, g_2, \dots, g_N)$. El movimiento del

enjambre se realiza en pasos temporales, que se traducen a nivel de algoritmo en iteraciones contiguas.

En cada iteración del método, k , cada una de las partículas de la población recorre el espacio de soluciones con una velocidad V_i hacia nuevas posiciones X_i , de acuerdo con su propia experiencia P_i , y con la experiencia aportada por el mejor de sus convecinos, G . En las primeras versiones del algoritmo (J. Kennedy, Particle swarm optimization, December 1995), esta formulación se reduce a las ecuaciones 5 y 6.

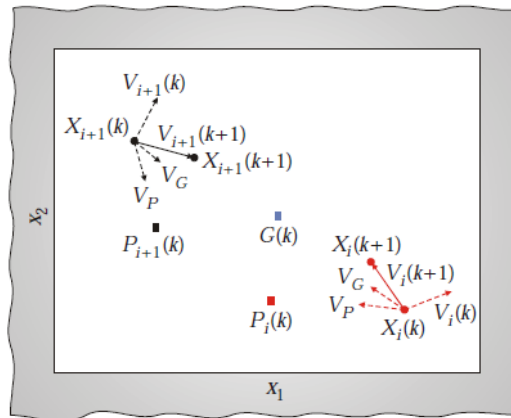
$$v_{in}(k + 1) = v_{in}(k) + c_1 r_1(k) \cdot [p_{in}(k) - x_{in}(k)] + c_2 r_2(k) \cdot [g_n(k) - x_{in}(k)] \quad ()$$

$$x_{in}(k + 1) = x_{in}(k) + v_{in}(k + 1) \cdot \Delta t \quad ()$$

A modo de resumen, el PSO se puede definir como un método de optimización rápido, fácil de implementar y eficaz, en el cual los parámetros a sintonizar incluyen el peso inercial, las constantes de aceleración c_1 y c_2 , el tamaño de la población y el límite superior de la velocidad v_{max} ; sin entrar a valorar aún la influencia del tipo de topología de la población, la importancia determinante de la función de fitness o la inserción de técnicas alternativas para mitigar el riesgo de convergencia prematura inherente a PSO. En definitiva, la formulación de PSO se reduce a caracterizar el movimiento de las partículas en base a un operador velocidad que debe aunar exploración y convergencia, descomponiendo para ello la velocidad en tres componentes en un intento por sintetizar el comportamiento social descrito en la Figura 2, tal y como se muestra en la Figura 4, (Pérez López, Octubre de 2005). Se debe tener en cuenta el control que se le da a la velocidad puesto que si esta es muy grande puede que ignore la solución global al pasar y

pasar por ella pero no poderla enfocar, para ello el límite de velocidad anteriormente nombrado v_{max} .

Figura . Componentes de vector velocidad. Movimiento sobre un espacio bidimensional de las partículas i e $i+1$ en la iteración $k+1$.



Fuente: Pérez López, J. R. (Octubre de 2005). Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo.

2.3.2.5 Selección de parámetros

Como en cualquier otro método de cómputo evolutivo, en PSO el ajuste de los parámetros condiciona el rendimiento posterior del algoritmo de optimización. Aunque existen múltiples estudios que intentan generalizar la selección de estos parámetros, dicha selección está íntimamente ligada con la naturaleza del problema a optimizar, siendo preciso lograr un balance óptimo entre exploración y convergencia. Al margen de los parámetros intrínsecos del algoritmo, debe resaltarse la importancia que adquiere la función de fitness. La utilización de una función de fitness que no introduzca una métrica adecuada para pesar la bondad de cada partícula hará emerger carencias del PSO que erróneamente pueden

asociarse con el algoritmo en sí, aún siendo más propias de los defectos de la propia función de fitness (Pérez López, Octubre de 2005).

Otro parámetro que precisa ser seleccionado con extrema rigurosidad es el tamaño de la población. Poblaciones excesivamente grandes exploran minuciosamente el espacio de búsqueda, pero como contrapartida, el coste computacional asociado con el aumento del número de evaluaciones de la función de fitness puede resultar excesivamente elevado. Estudios paramétricos revelan que una población de alrededor de 30 agentes es suficiente para múltiples problemas, y típicamente se utilizan poblaciones que oscilan entre 10 y 50 partículas, o entre 100 y 200 partículas para abordar problemas de gran complejidad y que necesitan resultados muy precisos por su aplicación (Pérez López, Octubre de 2005).

2.3.3 Algoritmos genéticos (GA)

Durante los últimos treinta años se ha experimentado un creciente interés por la construcción de sistemas de resolución de problemas basados en el proceso de evolución natural. Entre las técnicas más novedosas aplicadas a la optimización estructural se encuentran los Algoritmos Genéticos (AG), los cuales nacen de la necesidad de contar con métodos más robustos que mantengan un buen balance entre eficiencia y confiabilidad y que puedan ser aplicados a una gran variedad de problemas con geometrías no triviales.

Los Algoritmos Genéticos (AG) son algoritmos de búsqueda basados en la mecánica de la selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de cadenas con un intercambio de información estructurado, aunque al azar para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas

humanas (Goldberg, 1989). En toda generación, un nuevo grupo de criaturas artificiales (cadenas) es creado usando pedazos (bits) y piezas más aptas del anterior individuo; en ocasiones las nuevas partes son probadas en buena medida. Si bien al azar, los Algoritmos Genéticos no son un camino aleatorio simple. Explotan de manera eficiente la información histórica para especular sobre nuevos puntos de registro con expectativas de un mejor rendimiento.

Los AG han sido desarrollados por John Holland, sus colegas y sus alumnos en la Universidad de Michigan. Los objetivos de su investigación han sido dos: (1) abstraer y explicar rigurosamente los procesos de adaptación de los sistemas naturales, y (2) diseñar sistemas artificiales de software que conservan los mecanismos importantes de los sistemas naturales. Este enfoque ha conducido a importantes descubrimientos en la ciencia de los sistemas naturales y artificiales. (Goldberg, 1989)

El tema central de la investigación con algoritmos genéticos ha sido la robustez, el equilibrio entre la eficiencia y la eficacia necesarias para sobrevivir en muchos ambientes diferentes. Las implicaciones de la robustez de los sistemas artificiales son múltiples. Si los sistemas artificiales pueden ser más robusto, los costos de rediseño pueden ser reducidos o eliminados. (Goldberg, 1989)

Los AG fueron inspirados en el principio de selección natural y sobrevivencia del más fuerte, establecido por Charles Darwin en su libro *The Origin of Species* en 1859. De acuerdo con este principio, en una población de individuos, aquellos con “buenas” características genéticas, presentan mayores posibilidades de sobrevivir y reproducirse, mientras que los individuos menos “aptos” tienden a desaparecer durante el proceso evolutivo.

El AG realiza una simulación de evolución biológica por medio de una búsqueda multidireccional en el espacio de posibles soluciones del problema. Generalmente mantiene constante un número de posibles soluciones del problema (población) y, en cada generación, la población es modificada de tal manera que las soluciones “buenas” pueden ser “reproducidas” y pasar a la siguiente generación, y las “malas” pueden ser descartadas. El AG generalmente usa reglas de transición probabilística para seleccionar algunas soluciones que van a ser reproducidas y otras que serán descartadas.

Un algoritmo genético realiza una búsqueda multidireccional. Esta búsqueda es más intensiva en las direcciones más prometedoras, pero se dedica también una parte del esfuerzo a considerar regiones en principio de menor interés. La población sufre una evolución simulada: en cada generación los individuos buenos se reproducen mientras que los malos se mueren; en este proceso se generan individuos nuevos que heredan algunas características de sus progenitores mientras que otras son propias de cada individuo. Así pues podemos decir que los algoritmos genéticos tratan de alcanzar un equilibrio entre dos objetivos aparentemente en conflicto: explotar las buenas cualidades de las potenciales soluciones y explorar el espacio de búsqueda

En la naturaleza, los individuos de una población compiten constantemente con otros por recursos tales como comida, agua y refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones.

Es bien conocido por los biólogos que el proceso de evolución natural tiene lugar en los cromosomas, que poseen la información genética de los individuos. Aunque

no se sabe aún de manera completa la información específica codificada en los cromosomas de un individuo, las siguientes propiedades generales son aceptadas por todos los especialistas en el tema:

- a) La evolución es un proceso que opera fundamentalmente sobre los cromosomas.
- b) El proceso de selección natural es el responsable de que aquellos individuos que tienen mejor adaptación al entorno se reproduzcan más que los poco adaptados.
- c) En la reproducción es donde tiene lugar la evolución. Los procesos de mutación producen cambios en los cromosomas de los hijos con respecto a los padres y mediante las combinaciones del material genético de los padres se puede crear en los hijos material genético diferente.
- d) La evolución biológica carece de memoria. Lo único que pretende es producir individuos que se adapten de manera óptima al entorno en el que viven.

Los AG se caracterizan por utilizar operadores de cruce en el proceso de alteración de la población. Teniendo en cuenta lo anterior, los conceptos básicos que se manejan en la teoría de los algoritmos genéticos son los siguientes:

- **Individuo o cromosoma:** Un individuo determina una potencial solución del problema que se pretende resolver mediante el algoritmo genético.
- **Población:** Conjunto de individuos con los que se trabaja en el algoritmo genético. En un algoritmo genético los individuos que constituyen la

población van cambiando pero generalmente el tamaño de la misma permanece constante.

- **Función fitness:** Se trata de una función evaluadora de la calidad de un individuo como solución a nuestro problema. Permite la ordenación de los individuos de la población en cuanto a bondad de los mismos.
- **Cruce:** es una de las operaciones fundamentales que intervienen en todo algoritmo genético. Como norma general se aplica después de un proceso de selección de dos individuos y consiste en una combinación de los mismos para obtener como resultado otros dos nuevos individuos.
- **Mutación:** Constituye otra operación fundamental en un algoritmo genético. En este caso se selecciona un individuo, el cual sufre una pequeña modificación aleatoria en su codificación obteniéndose otro individuo nuevo.

En líneas generales y teniendo en cuenta la estructura de un algoritmo evolutivo, la estrategia operativa de un algoritmo genético consiste en partir de una población inicial de individuos, generada aleatoriamente, cada uno de los cuales representa una posible solución del problema. Estos individuos se evalúan mediante una función (*fitness*) que indica la calidad de la solución o grado de adaptación del individuo al entorno. A partir de esta situación inicial se realizan una serie de iteraciones en cada una de las cuales se simula la creación de una nueva generación de individuos a partir de la generación anterior. Este proceso consiste en aplicar los operadores genéticos de selección, cruce y mutación sobre los individuos. Hasta llegar finalmente a una población que, si el algoritmo converge adecuadamente, estará compuesta por buenos individuos, siendo el mejor de estos la solución que ofrece el algoritmo. Un proceso de los llamados elitistas siempre mantendrá en la población al mejor individuo alcanzado hasta el momento.

2.3.3.1 Diferencias entre métodos tradicionales y los algoritmos genéticos

Podemos decir que los Algoritmos Genéticos difieren de los principales métodos tradicionales de optimización en cuatro puntos fundamentales:

1. Los A.G. trabajan con codificaciones de los puntos del espacio de búsqueda en lugar de los puntos propiamente dichos.
2. Los A.G. realizan la búsqueda a partir de una población de puntos en lugar de un solo punto.
3. Los A.G. no utilizan derivadas ni otras propiedades de la función objetivo, sino únicamente la propia función objetivo.
4. Los A.G. se rigen mediante reglas de transición probabilísticas, no determinísticas.

Los A.G. requieren que el conjunto de puntos del espacio de búsqueda estén codificados mediante cadenas finitas sobre un alfabeto finito. (Goldberg, 1989)

2.3.3.2 Representación

Antes de que un algoritmo genético pueda ponerse a trabajar en un problema, se necesita un método para codificar las soluciones potenciales del problema de forma que una computadora pueda procesarlas. La elección de una representación adecuada de las posibles soluciones de un problema es la base

para el uso de AG en la solución de problemas del mundo real y condicionan todas las etapas siguientes del algoritmo. Para cualquier aplicación se debe realizar un análisis cuidadoso para garantizar una representación adecuada de las soluciones y una selección de los operadores genéticos específicos para el problema, lo cual es significativo.

La representación de una variable puede ser hecha utilizando métodos diferentes, destacándose entre estos, la codificación binaria y la real o de lo contrario, las representaciones que son típicos de un problema particular y tratar de reflejar de alguna manera, la estructura natural de los datos problema. Por lo general, la representación binaria se utiliza para representar problemas con variables discretas y la representación real de los problemas con las variables continuas. (Castilho, Febrero de 2003)

2.3.3.3 Codificación binaria

Los trabajos realizados con AG usaban como codificación de cromosoma, la representación binaria. En este tipo de representación, los valores de las variables son codificados como cadenas de caracteres binarios (0 y 1). El tamaño de la cadena esta directamente ligado a los límites de los valores que esas variables pueden asumir.

En caso de una variable continua la determinación del numero de bits (m) para representar sus posibles valores depende del límite inferior (b) y superior (c), el intervalo de valores que la variable puede asumir, el dominio del problema y la precisión que la solución requerirá, dada por el parámetro k . Esto se realiza mediante la determinación del entero (m) que satisface la ecuación 7.

$$2^{m-1} \leq (c - b)10^k \leq 2^m - 1 \quad ()$$

Cuando se trata de una variable discreta, el conjunto de valores que esta asume puede ser representado como una lista. Por ejemplo la variable *área de acero comercial*. La cantidad de bits que se necesitan para representar esta variable están dados por la ecuación 8. (Castilho, Febrero de 2003)

$$2^{nb} = nv \quad ()$$

Donde:

nb: Número de bits

nv: Número de posibles valores asumidos

Desventajas De La Representación Binaria. Hay cierto tipo de problemas del mundo real que no se ajusta a la representación binaria.

Ejemplo: Optimización de una función con alta dimensionalidad (ej.: 50 variables) con una buena precisión (ej.: 5 decimales)

- El cromosoma puede ser muy largo.
- Los operadores genéticos deberán ser adaptados para cada tipo de problema.
- Epistasis: el valor de un bit puede suprimir las contribuciones de aptitud de otros bits en el cromosoma.
- Representación natural: problemas como el del viajero se prestan de manera natural para el uso de representaciones con mayor cardinalidad que la binaria (uso de permutaciones de nros. enteros decimales)
- Soluciones ilegales producidas por los operadores genéticos aplicados sobre una representación binaria.

2.3.3.4 Codificación real

La representación de un cromosoma con números reales es más fácil de entender por los seres humanos que mediante una cadena de bits. Por otra parte, la representación con números reales requiere menos memoria. Otra ventaja de la representación real es su versatilidad a la hora de crear nuevos operadores, y las variaciones de los operadores existentes. (Castilho, Febrero de 2003)

2.3.3.5 Operadores Genéticos: Selección, cruce y mutación

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo.

2.3.3.5.1 Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no.

Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto la selección de un individuo estará relacionada con su valor de ajuste. No se debe sin embargo eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea.

Una opción bastante común consiste en seleccionar el primero de los individuos participantes en el cruce mediante alguno de los métodos expuestos a continuación y el segundo de manera aleatoria.

- **Selección Por Ruleta**

Propuesto por De Jong, es posiblemente el método más utilizado desde los orígenes de los Algoritmos Genéticos (Blickle & Thiele, 1995).

A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente la población está ordenada en base al ajuste por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0..1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Es un método muy sencillo, pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$). Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez.

En mucha bibliografía se suele referenciar a este método con el nombre de Selección de Montecarlo.

- **Selección Por Torneo**

La idea principal de este método consiste en realizar la selección en base a comparaciones directas entre individuos. Existen dos versiones de selección mediante torneo:

- Determinísticas.
- Probabilística.

En la versión determinísticas se selecciona al azar un número p de individuos (generalmente se escoge $p=2$). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en el paso de selección del ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo $[0..1]$, si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más alto y en caso contrario el menos apto. Generalmente p toma valores en el rango $0,5 < p \leq 1$.

Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular es el elitismo global. Se trata de un torneo en el que participan todos los individuos de la población con lo cual la selección se vuelve totalmente determinísticas. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor

se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Existen muchos otros algoritmos de selección. Unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Algunos de estos algoritmos son muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc.

2.3.3.5.2 Cruce

Una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación. Tal y como se ha indicado anteriormente el cruce es una estrategia de reproducción sexual.

Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90%.

Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar). Por el contrario utilizando una estrategia no destructiva la descendencia pasaría a la siguiente generación únicamente si supera la bondad del ajuste de los padres (o de los individuos a reemplazar). La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los padres. Al compartir las características buenas de dos individuos, la descendencia, o al

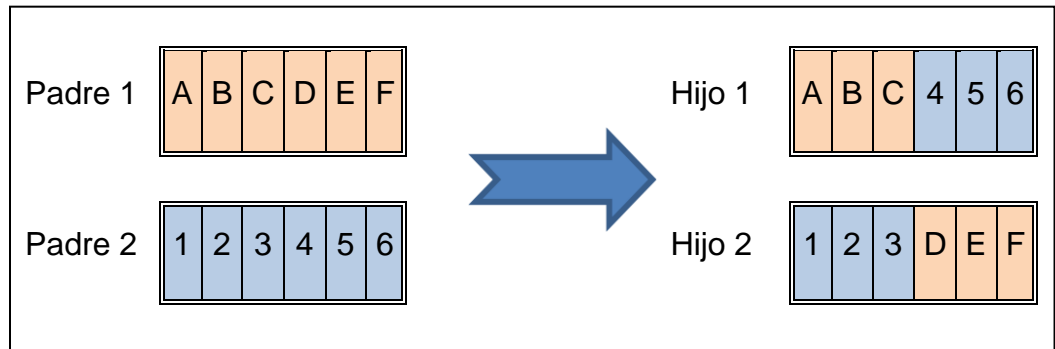
menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva garantizamos que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población - aunque dispersos y posiblemente levemente modificados por la mutación - en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente perdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación:

- Cruce de 1 punto.
- Cruce de 2 puntos.
- Cruce uniforme.
- **Cruce de 1 punto**

Es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres, ver figura 5. En la bibliografía suele referirse a este tipo de cruce con el nombre de SPX (Single Point Crossover).

Figura . Cruce de 1 Punto.



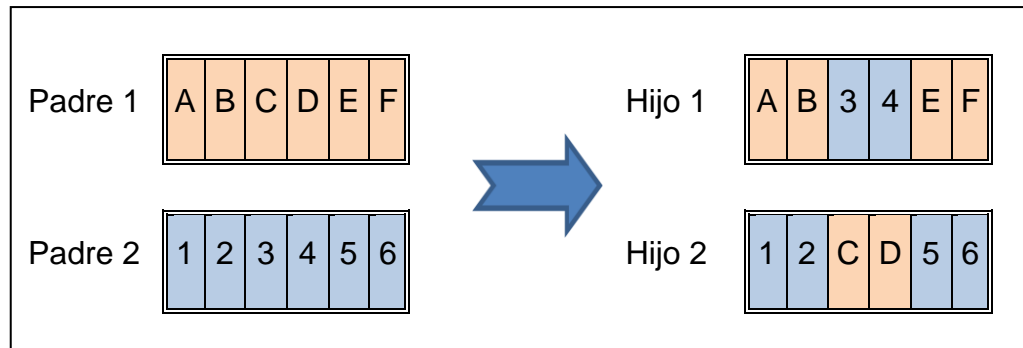
Fuente: Elaboración Propia

➤ Cruce De 2 Puntos

Se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres como en el caso anterior se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre (ver figura 6).

Generalmente se suele referir a este tipo de cruce con las siglas DPX (Double Point Crossover).

Figura . Cruce de 2 Punto.



Fuente: *Elaboración Propia*

➤ **Cruce Uniforme**

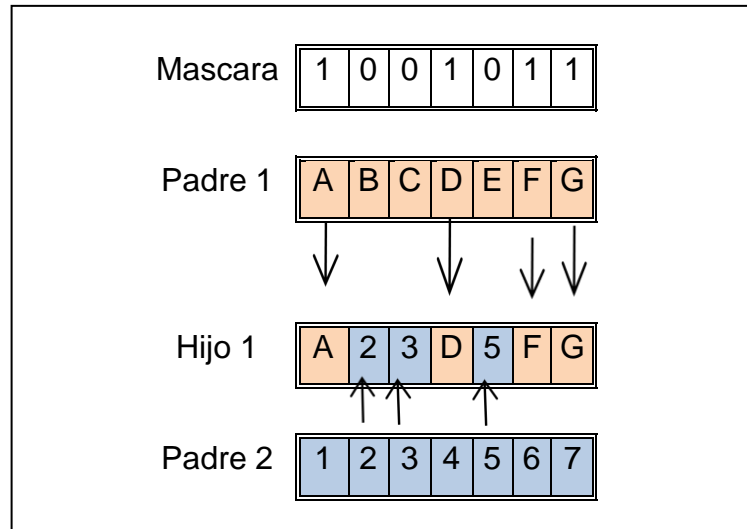
El cruce uniforme es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro padre.

Aunque se puede implementar de muy diversas formas, la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer padre. Si por el contrario hay un 0 el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

La descendencia contiene una mezcla de genes de cada uno de los padres, ver Figura 7. El número efectivo de puntos de cruce es fijo pero será por término medio $L/2$, siendo L la longitud del cromosoma (numero de alelos en representaciones binarias o de genes en otro tipo de representaciones).

Se suele referir a este tipo de cruce con las siglas UPX (Uniform Point Crossover).

Figura . Cruce Uniforme.



Fuente: Elaboración Propia

2.3.3.5.3 Algoritmos de Reemplazo

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población.

Existen diferentes métodos de reemplazo:

- Aleatorio: el nuevo individuo se inserta en un lugar cualquiera de la población.
- Reemplazo de padres: se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.

- Reemplazo de similares: una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.
- Reemplazo de los peores: de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

2.2.5.3.4 Copia

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reducirá notablemente y la búsqueda en el espacio del problema se focalizara en el entorno de ese individuo.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce, y si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

2.2.5.3.5 Mutación

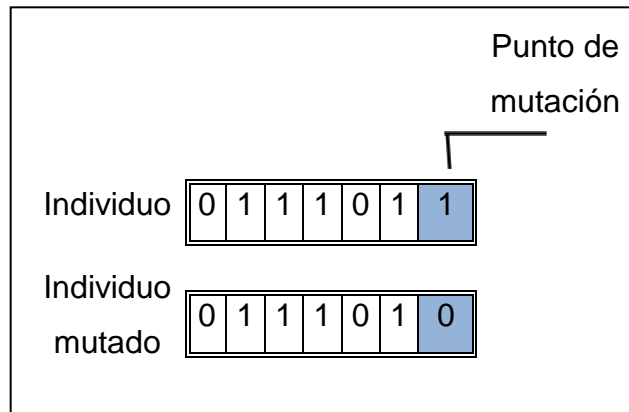
La mutación de un individuo provoca que alguno de sus genes, generalmente uno solo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se suele utilizar de manera conjunta con el operador de cruce.

La utilización de este operador tiene como objetivo la introducción y manutención de la diversidad genética de la población. De esta forma, la mutación asegura que la probabilidad de examinar cualquier punto del espacio no sea cero (0), y también evita el problemas de los mínimos locales. El operador de mutación se aplica a individuos con una probabilidad dada por la tasa de mutación p_m ($0,001 \leq p_m \leq 0,1$). La mayoría de veces la mutación es aplicada después de la operación de cruce. Si p_m es muy baja puede haber un compromiso con la diversidad de población. Si p_m es muy alta, sucederán muchas perturbaciones aleatorias y los hijos probablemente comenzarán a perder sus similitudes con los padres y pueden comprometer la convergencia.

Generalmente en los AG el operador de mutación es tratado como un operador secundario con la función de recuperar el material genético perdido. Suponiendo que en una determinada generación todos los cromosomas de la población tenían una determinada posición, un dígito 0 y suponiendo que la solución óptima tenga un dígito 1 en aquella misma posición. Un operador de cruce no puede regenerar un 1 en aquella posición, mientras que con el operador de mutación existe una oportunidad que el bit 0 pase a ser 1. Figura 8.

Figura . Mutación.



Fuente: Elaboración Propia

En resumen, para resolver un problema particular mediante un algoritmo genético, debemos tener las siguientes componentes:

- Una codificación o representación genética de las soluciones potenciales del problema (cadenas binarias en el caso del algoritmo genético simple).
- un modo de crear la población inicial (aleatoria en el algoritmo genético simple).
- una función de evaluación que juega el papel del entorno permitiendo clasificar a los individuos en términos de su fitness,
- operadores genéticos de reproducción, cruce y mutación que simulan la evolución de las poblaciones, y
- valores de los parámetros del algoritmo: longitud de los individuos, tamaño de la población, número de generaciones y probabilidad de mutación.

2.3.4 Otras metodologías

Los métodos heurísticos que se nombraran a continuación son Random ascent (RA), Hero, Simulated annealing (SA) y Tabu search (TS), siendo entre estos el SA el método heurístico más utilizado en la resolución de problemas combinatorios, mientras que Hero ha sido muy utilizado en la resolución de problemas de planificación forestal formulados mediante funciones de utilidad multiatributo (MAUT)². RA es utilizado como método referencia ya que debido a su proceso de búsqueda cualquier otro método heurístico debería resultar más efectivo que este, se nombra en primera instancia Ant Colony Optimization puesto que resulta de gran importancia por su similitud con los PSO representando comportamientos de especies determinadas en este caso las hormigas y aprovechando su socialización para encontrar mejores resultados. Los anteriores métodos usados se describen brevemente a continuación.

2.3.4.1 Optimización Mediante Colonias De Hormigas (ACO)

Los algoritmos ACO son modelos inspirados en el comportamiento de colonias de hormigas reales. Estudios realizados explican cómo animales casi ciegos, como son las hormigas, son capaces de seguir la ruta más corta en su camino de ida y vuelta entre la colonia y una fuente de abastecimiento. Esto es debido a que las hormigas pueden transmitirse información entre ellas gracias a que cada una de ellas, al desplazarse, va dejando un rastro de una sustancia llamada feromona a lo largo del camino seguido. Así, mientras una hormiga aislada se mueve de forma esencialmente aleatoria, los "agentes" de una colonia de hormigas detectan el rastro de feromona dejado por otras hormigas y tienden a seguir dicho rastro. Éstas a su vez van dejando su propia feromona a lo largo del camino recorrido y

² Para ampliar ver Pukkala, T. y Kangas, J.; 1993. A heuristic optimization method for forest planning and decision making.

por tanto lo hacen más atractivo, puesto que se ha reforzado el rastro de feromona.

Sin embargo, la feromona también se va evaporando con el paso del tiempo provocando que el rastro de feromona sufra, por otro lado, cierto debilitamiento. En definitiva, puede decirse que el proceso se caracteriza por una retroalimentación positiva, en la que la probabilidad con la que una hormiga escoge un camino aumenta con el número de hormigas que previamente hayan elegido el mismo camino.

El primer algoritmo basado en la optimización mediante colonias de hormigas fue aplicado al Problema del Viajante, obteniéndose unos resultados bastante alentadores (Salazar Pinto, 2009). A partir de dicho algoritmo se han desarrollado diversos heurísticos que incluyen varias mejoras, y han sido aplicados no solo al TSP sino también a problemas como el VRP y el QAP entre otros (Dorigo & Di Caro, 1999).

Los algoritmos ACO son procesos iterativos. En cada iteración se "lanza" una colonia de m hormigas y cada una de las hormigas de la colonia construye una solución al problema. Las hormigas construyen las soluciones de manera probabilística, guiándose por un rastro de feromona artificial y por una información calculada a priori de manera heurística.

2.3.4.2 Random ascent (RA)

En RA se produce una solución inicial asignando aleatoriamente un régimen selvícola a cada unidad mínima de gestión. A continuación, una unidad mínima de gestión y un nuevo régimen selvícola para este son seleccionados aleatoriamente. En caso de que el nuevo régimen mejore el valor de la función objetivo, este es

incluido en la solución. Este proceso de búsqueda se repite hasta que un número máximo de iteraciones es alcanzado.

2.3.4.3 Hero³

En Hero en primer lugar, un régimen selvícola es seleccionado aleatoriamente para cada unidad mínima de gestión, produciendo así una solución inicial. En segundo lugar, de uno en uno cada una de las unidades de gestión son examinadas secuencialmente con el objeto de comprobar si otro régimen selvícola mejoraría el valor de la función objetivo. En caso afirmativo, el régimen que hace mejorar la función objetivo substituye al previo.

2.3.4.4 Simulated annealing (SA)⁴

En SA se produce una solución inicial asignando aleatoriamente un régimen selvícola para cada unidad mínima de gestión. A continuación, una unidad mínima de gestión y un nuevo régimen selvícola para esta son seleccionados aleatoriamente. Con el objeto de evitar una convergencia prematura en un óptimo local, SA, además de aceptar todo movimiento de mejora (de la función objetivo), acepta también movimientos de no mejora de acuerdo con unas probabilidades.

2.3.4.5 Tabu search (TS)⁵

La característica principal de TS es el uso de una lista tabú la cual condiciona el proceso de búsqueda, por ejemplo mediante la prohibición de repetir movimientos

³ Para ampliar conceptos ver Pukkala, T. y Kangas, J.; 1993. A heuristic optimization method for forest planning and decision making.

⁴ Más detalles en Dowsland, K.A.; 1993. Simmualted annealing.

⁵ Más detalles en Glover, F. y Laguna, M.; 1993. Tabu search.

recientes. Como en los métodos previos, TS comienza su búsqueda a partir de una solución inicial aleatoria. A continuación, de entre los movimientos alternativos se selecciona el movimiento no-tabú que produzca la mejor solución.

La búsqueda Tabú, de naturaleza determinista, tiene capacidad para escapar de los mínimos o máximos locales, aprovechando un cierto conocimiento acerca del dominio de búsqueda y actualizando la solución en curso con el mejor punto de su vecindad. Por otra parte, dentro de los métodos heurísticos de naturaleza probabilística, denominación asociada con el hecho de que la optimización depende de eventos aleatorios, existen dos familias diferenciadas: aquellos que utilizan un único punto de partida y aquellos que utilizan una población (Pérez López, Octubre de 2005).

3. INVESTIGACIÓN

3.1 DISEÑO

El diseño de los elementos de la cercha se hace de acuerdo a las restricciones del Reglamento Colombiano De Construcción Sismo Resistente (NSR-10) en su Título F (Estructuras Metálicas).

3.1.1 Definición de cargas

Se tuvieron en cuenta las cargas muertas de la estructuras e incluso las transmitidas a la cercha por análisis previos. De igual forma se tuvieron en cuenta las cargas vivas, aquellas cargas producidas por el uso y ocupación de la edificación y que no deben incluir cargas ambientales tales como viento y sismo, y los efectos producidos por el viento y las cargas de sismo -carga de sismo es la efectuada por un sismo sinusoidal efectuada en la estructura a estudiar-.

Teniendo en cuenta que este proyecto quiere comparar los resultados obtenidos con los ya publicados en el artículo Vedat Toğan, A. T. (18 January 2006). Optimization of 3d trusses with adaptive approach in genetic algorithms. *Engineering Structures* 28 (2006) , 1019–1027, para así tener una base aprobada y sustentada para la comprobación del estudio realizado, se optó por hacer las siguientes consideraciones:

- Las cargas que el artículo representa como obtenidas por análisis previos de la cubierta, comprende, el peso muerto de la misma y los efectos

producidos por el viento y las cargas vivas representativas en la misma, todas ya mayoradas.

- El análisis por carga muerta se realizó por distribución simétrica en cada elemento, con esto se quiere decir, el peso que efectúa el elemento es repartido por partes iguales en los nodos en los que se encuentra apoyado.
- Solo se optó por mayorar la carga muerta producida por los elementos de la estructura por un factor de 1.2 como lo expresa la norma NSR10.

NOTA: No se especifica una combinación de carga puesto que la cercha ya tiene un análisis previo, las cargas de análisis previos para el estudio de la cercha se muestran en el apartado 3.1.4 , llamado datos de la estructura, del tercer capítulo.

3.1.2 Resistencia de diseño a tensión

La resistencia de diseño para miembros solicitados por tensión $\phi_t P_n$, se tomará como el menor valor entre los obtenidos para los estados límites de fluencia por tensión sobre el área bruta y rotura por tensión sobre el área neta.

a) Para fluencia por tensión sobre el área bruta:

$$P_n = F_y A_g \quad ()$$

$$\phi_t = 0,9 \quad ()$$

b) Para rotura por tensión sobre el área neta:

$$P_n = F_u A_e \quad ()$$

$$\phi_t = 0,75 \quad ()$$

Donde:

A_e = Área neta efectiva.

A_g = Área bruta del miembro.

F_y = Esfuerzo de fluencia mínimo especificado para el tipo de acero usado.

F_u = resistencia a tensión mínima especificada para el tipo de acero usado.

3.1.3 Resistencia de diseño a compresión

La resistencia de diseño a compresión se tomará igual a $\phi_c P_n$, donde: $\phi_c = 0,9$.

Pandeo por flexión de miembros de miembros sin elementos esbeltos: La resistencia nominal a compresión P_n , para el estado límite de pandeo por flexión, se tomará igual a:

$$P_n = F_{cr} A_e \quad ()$$

Donde el esfuerzo de pandeo por flexión F_{cr} , se calcula como sigue:

$$\text{Cuando } \frac{KL}{r} \leq 4,71 \sqrt{\frac{E}{F_y}} \text{ ó } (F_e \geq 0,44F_y) \quad ()$$

$$F_{cr} = [0,658^{F_y/F_e}] F_y$$

$$\text{Cuando } \frac{KL}{r} > 4,71 \sqrt{\frac{E}{F_y}} \text{ ó } (F_e < 0,44F_y) \quad ()$$

$$F_{cr} = 0,877 F_e$$

$$F_e = \frac{\pi^2 E}{(KL/r)^2}$$

Miembros con elementos esbeltos:

$$P_n = F_{cr} A_e$$

Donde F_{cr} , se calcula como sigue:

$$\text{Cuando } \frac{KL}{r} \leq 4,71 \sqrt{\frac{E}{QF_y}} \text{ ó } (F_e \geq 0,44QF_y) \quad ()$$

$$F_{cr} = Q[0,658^{QF_y/F_e}]F_y$$

$$\text{Cuando } \frac{KL}{r} > 4,71 \sqrt{\frac{E}{QF_y}} \text{ ó } (F_e < 0,44QF_y) \quad ()$$

$$F_{cr} = 0,877F_e$$

Para secciones circulares cargadas axialmente, cuando $0,11 \frac{E}{F_y} < \frac{D}{t} < \frac{E}{F_y}$:

$$Q = \frac{0,038E}{F_y(D/t)} + \frac{2}{3} \quad ()$$

Donde:

D = Diámetro exterior

t = Espesor de pared

3.1.4 Datos de la estructura

La cercha en 3D en la Fig. 10, está conformada por 200 miembros. Los perfiles utilizados en la estructura en estudio son perfiles circulares huecos. Los esfuerzos a tensión y compresión admisibles de cada miembro deben cumplir las especificaciones del Reglamento Colombiano de Construcción Sismo Resistente (NSR-10) en su Título F (Estructuras Metálicas). Los nodos de la cuerda superior

de la estructura están sujetos a una carga vertical de 13,5 KN, y los desplazamientos esta restringidos a 20 mm.

La optimización se hará para dos casos, dependiendo de la agrupación de miembros que se tome, el primer caso se hará en base a la a agrupación de miembros descritos en la literatura (Vedat Toğan, 18 January 2006) y el segundo caso se hará para una agrupación de miembros propuesta por los autores.

Caso1: Los miembros se dividen en tres grupos de acuerdo al área de su sección transversal. En el primer grupo se encuentran los primeros 60 miembros que corresponden a aquellos elementos en la parte inferior de la estructura, el segundo grupo los siguiente 100 miembros es decir del elemento 61 hasta el 160 que corresponden a los miembros diagonales y por último el tercer grupo lo conforman los restantes miembros que van desde el elemento 161 hasta el 200 que corresponde a los miembros de la parte superior de la estructura.

Caso 2: Los miembros se dividen en cuatro grupos. Al igual que el caso 1 el primer grupo corresponde a los primeros 60 miembros de la estructura, el segundo grupo reúne los elementos diagonales desde el elemento 65 hasta el 160, es decir tenemos 96 miembros en este grupo, el tercer grupo está conformado igual que en el caso 1, los elementos desde el miembro 161 hasta el 200, y por ultimo un nuevo grupo que lo conforman los 4 elementos diagonales de las cuatro esquinas de la estructura que no fueron incluidos en el grupo dos, estos son los miembros 61, 62, 63 y 64.

Las siguientes figuras (9 y 10) son basadas en la cercha del articulo *Vedat Toğan, A. T. (18 January 2006). Optimization of 3d trusses with adaptive approach in genetic algorithms. Engineering Structures 28 (2006) , 1019–1027.*

Figura Cercha en estudio, planta y perfil.

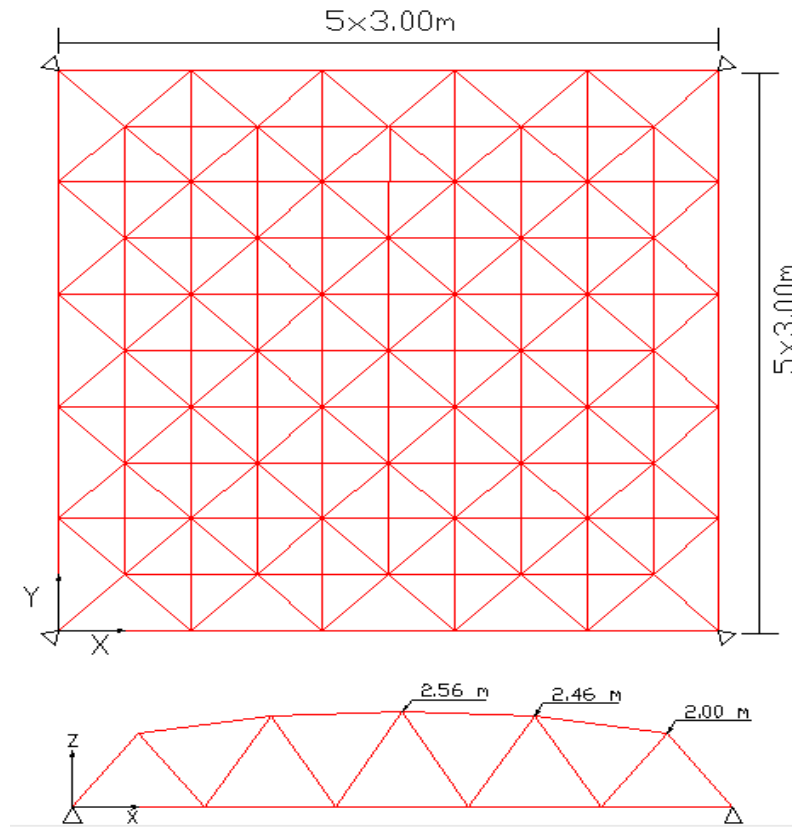


Figura . Cercha en estudio, vista 3D.

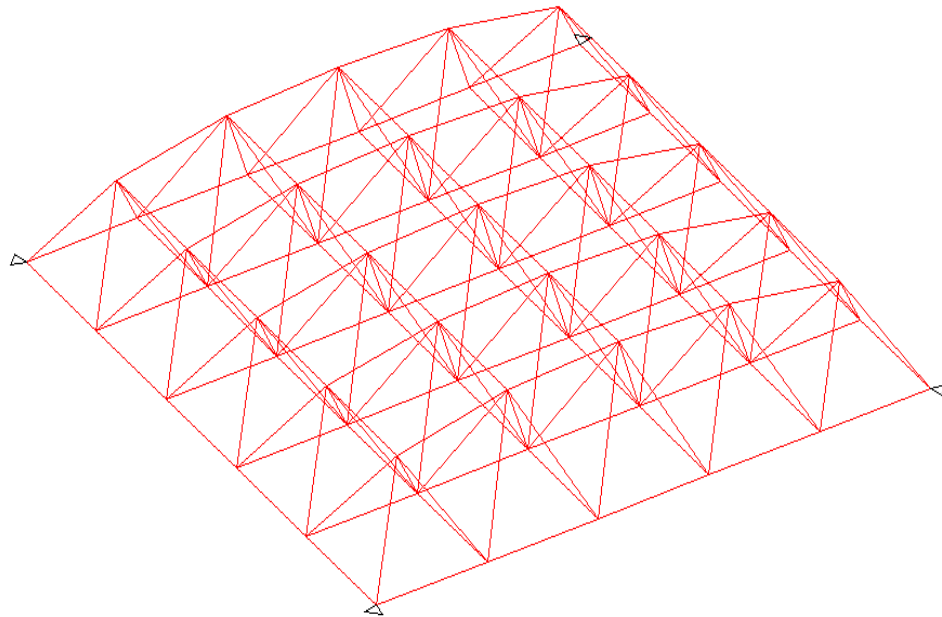


Tabla Perfiles utilizados en el diseño de la cercha

D exterior [cm]	Espesor [mm]	Área [cm ²]	Peso [kg/m]	R de giro [cm]
2,07	2,5	1,43	1,12	0,65
2,61	2,5	1,85	1,46	0,84
3,29	2	1,94	1,53	1,11
3,29	2,5	2,39	1,88	1,1
4,16	2,5	3,07	2,41	1,39
4,79	2,5	3,56	2,8	1,61
4,22	3	3,7	2,9	1,39
4,83	3	4,27	3,35	1,6
7,24	2	4,42	3,47	2,49
5,99	2,5	4,51	3,54	2,03
6,03	3	5,4	4,24	2,03
7,24	2,5	5,49	4,31	2,47
8,82	2	5,5	4,32	3,03
7,24	3	6,54	5,13	2,46
8,82	2,5	6,73	5,29	3,03
6,03	4	7,08	5,56	2
11,35	2	7,11	5,58	3,91
8,82	3	8,03	6,31	3,02
7,24	4	8,59	6,75	2,42
11,35	2,5	8,72	6,84	3,93
11,35	3	10,41	8,17	3,91
8,82	4	10,59	8,31	2,98

7,3	5	10,69	8,39	2,41
11,35	4	13,76	10,8	3,87
8,89	5,5	14,41	11,31	2,96
12,7	4	15,46	12,1	4,35
8,89	6,35	16,47	12,93	2,93
11,43	6	20,41	16,03	3,83
16,83	5	25,65	20,14	5,78
16,83	6	30,59	24,02	5,74
21,91	5	33,63	26,4	7,57
16,83	7,11	36,01	28,27	5,7

Fuente: Tubos COLMENA (COLMENA Acero en evolución)

En la tabla 1 se muestran los perfiles utilizados para el diseño de la cercha son perfiles tubulares Colmena con $F_y=322$ MPa.

La tabla 2 muestra las coordenadas de los nodos de la cercha, los nodos 58, 59, 60 y 61 son los nodos restringidos, los demás son nodos libres. En la tabla 3 se muestra la enumeración de los elementos y sus respectivos nodo inicial “ n_i ” y nodo final “ n_f ”.

Tabla Coordenadas de los nodos

	COORDENADAS				COORDENADAS		
Nodo	X	Y	Z	Nodo	X	Y	Z
1	3	0	0	32	12	15	0
2	6	0	0	33	1,5	1,5	2
3	9	0	0	34	4,5	1,5	2,46
4	12	0	0	35	7,5	1,5	2,56
5	0	3	0	36	10,5	1,5	2,46
6	3	3	0	37	13,5	1,5	2
7	6	3	0	38	1,5	4,5	2
8	9	3	0	39	4,5	4,5	2,46
9	12	3	0	40	7,5	4,5	2,56
10	15	3	0	41	10,5	4,5	2,46
11	0	6	0	42	13,5	4,5	2
12	3	6	0	43	1,5	7,5	2
13	6	6	0	44	4,5	7,5	2,46
14	9	6	0	45	7,5	7,5	2,56
15	12	6	0	46	10,5	7,5	2,46
16	15	6	0	47	13,5	7,5	2
17	0	9	0	48	1,5	10,5	2
18	3	9	0	49	4,5	10,5	2,46
19	6	9	0	50	7,5	10,5	2,56
20	9	9	0	51	10,5	10,5	2,46
21	12	9	0	52	13,5	10,5	2
22	15	9	0	53	1,5	13,5	2
23	0	12	0	54	4,5	13,5	2,46
24	3	12	0	55	7,5	13,5	2,56

25	6	12	0	56	10,5	13,5	2,46
26	9	12	0	57	13,5	13,5	2
27	12	12	0	58	0	0	0
28	15	12	0	59	15	0	0
29	3	15	0	60	0	15	0
30	6	15	0	61	15	15	0
31	9	15	0				

Fuente: *Elaboración Propia*

Tabla Enumeración de los miembros

Miembro	ni	nf	Miembro	ni	nf	Miembro	ni	nf
1	1	2	71	4	36	141	25	54
2	2	3	72	4	37	142	25	55
3	3	4	73	5	33	143	26	50
4	5	6	74	5	38	144	26	51
5	6	7	75	6	33	145	26	55
6	7	8	76	6	34	146	26	56
7	8	9	77	6	38	147	27	51
8	9	10	78	6	39	148	27	52
9	11	12	79	7	34	149	27	56
10	12	13	80	7	35	150	27	57
11	13	14	81	7	39	151	28	52
12	14	15	82	7	40	152	28	57
13	15	16	83	8	35	153	29	53
14	17	18	84	8	36	154	29	54
15	18	19	85	8	40	155	30	54

16	19	20	86	8	41	156	30	55
17	20	21	87	9	36	157	31	55
18	21	22	88	9	37	158	31	56
19	23	24	89	9	41	159	32	56
20	24	25	90	9	42	160	32	57
21	25	26	91	10	37	161	33	34
22	26	27	92	10	42	162	34	35
23	27	28	93	11	38	163	35	36
24	29	30	94	11	43	164	36	37
25	30	31	95	12	38	165	38	39
26	31	32	96	12	39	166	39	40
27	1	58	97	12	43	167	40	41
28	4	59	98	12	44	168	41	42
29	29	60	99	13	39	169	43	44
30	32	61	100	13	40	170	44	45
31	1	6	101	13	44	171	45	46
32	2	7	102	13	45	172	46	47
33	3	8	103	14	40	173	48	49
34	4	9	104	14	41	174	49	50
35	5	11	105	14	45	175	50	51
36	6	12	106	14	46	176	51	52
37	7	13	107	15	41	177	53	54
38	8	14	108	15	42	178	54	55
39	9	15	109	15	46	179	55	56
40	10	16	110	15	47	180	56	57
41	11	17	111	16	42	181	33	38
42	12	18	112	16	47	182	34	39
43	13	19	113	17	43	183	35	40

44	14	20	114	17	48	184	36	41
45	15	21	115	18	43	185	37	42
46	16	22	116	18	44	186	38	43
47	17	23	117	18	48	187	39	44
48	18	24	118	18	49	188	40	45
49	19	25	119	19	44	189	41	46
50	20	26	120	19	45	190	42	47
51	21	27	121	19	49	191	43	48
52	22	28	122	19	50	192	44	49
53	24	29	123	20	45	193	45	50
54	25	30	124	20	46	194	46	51
55	26	31	125	20	50	195	47	52
56	27	32	126	20	51	196	48	53
57	5	58	127	21	46	197	49	54
58	10	59	128	21	47	198	50	55
59	23	60	129	21	51	199	51	56
60	28	61	130	21	52	200	52	57
61	33	58	131	22	47			
62	37	59	132	22	52			
63	53	60	133	23	48			
64	57	61	134	23	53			
65	1	33	135	24	48			
66	1	34	136	24	49			
67	2	34	137	24	53			
68	2	35	138	24	54			
69	3	35	139	25	49			
70	3	36	140	25	50			

Fuente: Elaboración Propia

3.1.5 Modelo matemático para problemas de optimización estructural con variables discretas⁶

Un problema de optimización del diseño estructural con variables discretas puede ser formulado como un problema de programación no lineal. En la optimización del peso de una cercha, las áreas de la sección transversal de los miembros de la armadura son seleccionadas como las variables de diseño. Cada una de las variables de diseño se selecciona de una lista de secciones discretas basadas en producción estándar. La función objetivo es el peso de la estructura. El diseño de las secciones transversales también deberá cumplir algunas restricciones. El problema de optimización de diseño para las variables discretas se puede expresar de la siguiente manera:

$$\begin{aligned} &\text{Minimizar } f(x^1, x^2, \dots, x^d), \quad d = 1, 2, \dots, D \\ &\text{Sujeto a: } g_q(x^1, x^2, \dots, x^d), \leq 0, \quad d = 1, 2, \dots, D \\ &\quad q = 1, 2, \dots, M \\ &\quad x^d \in S_d = \{X_1, X_2, \dots, X_p\} \end{aligned}$$

Donde $f(x^1, x^2, \dots, x^d)$ es la función del peso de la estructura, que es una función escalar. Y x^1, x^2, \dots, x^d representa un grupo de variables de diseño. La variable de diseño x^d pertenece a un escalar S_d , que incluye todas las variables discretas permisivas $\{X_1, X_2, \dots, X_p\}$. La desigualdad $g_q(x^1, x^2, \dots, x^d), \leq 0$ representa la función de restricciones. D y M son el número de variables de diseño y funciones de desigualdad, respectivamente. “ p ” representa el número de variables disponibles.

⁶ (L.J. Li, 18 February 2009)

3.2 OPTIMIZACIÓN

3.2.1 Optimización del peso de una cercha 3d mediante los algoritmos PSO y GA

A continuación se hace una descripción del algoritmo implementado en código matlab® para la solución estática y dinámica de la cercha en estudio:

1. **Análisis estático:** para realizar un análisis estático de las fuerzas internas que soportan los miembros de la estructura, y determinar los desplazamientos en los nodos de la cercha, se empleó el análisis matricial.
2. **Análisis dinámico:** el método empleado para análisis de fuerzas dinámicas fue el de integración directa propuesto por Newmark.

Enfatizando en el **análisis dinámico** se muestra la ecuación general que gobierna el análisis dinámico en una estructura es la siguiente:

$$M\ddot{U} + C\dot{U} + KU = R \quad ()$$

Donde **M**, **C** y **K** son las matrices de masa, amortiguamiento y rigidez respectivamente; R es el vector de cargas externas; y **U**, **U** y **U** son vectores de desplazamientos, velocidades y aceleraciones respectivamente.

Para realizar el análisis de las fuerzas sísmicas que actúan sobre la estructura, empleamos el método de integración directa de Newmark, explicado a continuación.

Son usadas las siguientes suposiciones:

$$\dot{U}_{t+1} = \dot{U}_t + [(1 - \delta)\ddot{U}_t + \delta\ddot{U}_{t+1}]\Delta t \quad ()$$

$$U_{t+1} = U_t + \dot{U}_t\Delta t + \left[\left(\frac{1}{2} - \alpha\right)\ddot{U}_t + \alpha\ddot{U}_{t+1}\right]\Delta t^2 \quad ()$$

Donde α y δ son parámetros que pueden ser determinados por la precisión y estabilidad de la integración. Newmark propone en su regla trapezoidal que, $\alpha=0,25$ y $\beta=0,5$.

En adición a (20) y (21), para la solución de los desplazamientos, velocidades y aceleraciones en el instante de tiempo $(t+1)$, la ecuación de equilibrio (19) en el instante $(t+1)$, es considerada:

$$M\ddot{U}_{t+1} + C\dot{U}_{t+1} + KU_{t+1} = R_{t+1} \quad ()$$

Resolviendo de (21) para \ddot{U}_{t+1} en términos de U_{t+1} y luego sustituyendo para \ddot{U}_{t+1} en (20), se obtiene ecuaciones para \ddot{U}_{t+1} y \dot{U}_{t+1} , cada una en términos U_{t+1} únicamente. Estas dos relaciones para \ddot{U}_{t+1} y \dot{U}_{t+1} , son sustituidas en (22) para encontrar U_{t+1} , después usando (20) y (23), \ddot{U}_{t+1} y \dot{U}_{t+1} pueden ser calculados.

A continuación el método de integración de Newmark paso a paso:

A. Cálculos iniciales:

1. Formar la matriz de rigidez **K**, matriz de masa **M**, y la matriz de amortiguamiento **C**.
2. Establecer las condiciones iniciales $\ddot{U}_0, \dot{U}_0, U_0$.
3. Seleccionar Δt , los parámetros α y δ , y calcular las constantes de integración:

$$a_0 = \frac{1}{\alpha\Delta t^2}; \quad a_1 = \frac{\delta}{\alpha\Delta t}; \quad a_2 = \frac{1}{\alpha\Delta t}; \quad a_3 = \frac{1}{2\alpha} - 1; \quad a_4 = \frac{\delta}{\alpha} - 1;$$

$$a_5 = \frac{\Delta t}{2} \left(\frac{\delta}{\alpha} - 2 \right); \quad a_6 = \Delta t(1 - \delta); \quad a_7 = \delta \Delta t;$$

4. Formar la matriz de rigidez efectiva \hat{K} : $\hat{K} = K + a_0 M + a_0 C$

B. Para cada instante de tiempo:

Calcular las cargas efectivas en $(t+\Delta t)$:

$$\hat{R}_{t+1} = R_{t+1} + M(a_0 U_t + a_2 \dot{U}_t + a_3 \ddot{U}_t) + C(a_1 U_t + a_4 \dot{U}_t + a_5 \ddot{U}_t) \quad ()$$

Resolver los desplazamientos en $(t+\Delta t)$:

$$\hat{K} U_{t+1} = \hat{R}_{t+1} \quad ()$$

Calcular aceleraciones y velocidades en $(t+\Delta t)$:

$$\ddot{U}_{t+1} = a_0 (U_{t+1} - U_t) - a_2 U_t - a_3 \ddot{U}_t \quad ()$$

$$\dot{U}_{t+1} = \dot{U}_t + a_6 \ddot{U}_t + a_7 \ddot{U}_{t+1} \quad ()$$

No se consideró el amortiguamiento para el análisis de la estructura.

Antes de hacer la optimización ya sea por el algoritmo “PSO” o por “AG” se deben ingresar los siguientes datos para el análisis de la cercha:

- En una matriz “ge” de tamaño (b) filas por 9 columnas, se guardan los datos de la geometría de la cercha, donde (b) es el número de miembros de la cercha, el orden de las columnas es el siguiente: en la tres primeras columnas tenemos: Elemento (b), nodo inicial, y nodo final

respectivamente, las siguientes tres columnas corresponden a las coordenadas iniciales en X, Y, y en Z del elemento respectivamente, y por ultimo tres columnas que corresponden a las coordenadas finales en X, Y, y en Z de elemento respectivamente.

- Se debe definir una matriz “fr” de tamaño (t) filas por 7 columnas, donde se guardan los datos de las fuerzas externas y restricciones en los nodos, (t) es el numero nodos en la cercha. La distribución de los datos en las columnas es la siguiente: la primera columna almacena el nodo (t), las siguientes tres columnas guarda las fuerzas en X, Y, y en Z a las que está sometido el nodo respectivamente y las últimas tres columnas guarda las restricciones al movimiento en X, Y, y en Z del nodo respectivamente (“1” si tiene restricción y “0” si es un nodo libre).
- El valor de “E” almacena el modulo de elasticidad del acero en KN/m².
- Para el análisis sísmico se crea la matriz “acel” que en la primera columna guarda los intervalos de tiempo y la segunda columna guarda las aceleraciones en cada instante de tiempo.
- El valor guardado en “int” equivale al número de intervalos de tiempo en que se divida la duración del sismo y el valor de “dt” guarda valor del delta de tiempo.
- Se creó una matriz llamada “Perfiles” donde se almacena la información que muestra la tabla 1.

3.2.2 Descripción del algoritmo “pso” implementado en el programa Matlab® para la optimización del peso de la cercha.

Cuando se inicia el algoritmo “PSO”, este algoritmo pide al usuario que ingrese algunos parámetros para su funcionamiento, estos parámetros son escogidos de acuerdo a la complejidad del problema que se quiere resolver.

3.2.2.1 Algoritmo “pso” para variables discretas

La optimización con variables discretas es una combinación de problemas de optimización que obtienen la mejor solución de todas las posibles combinaciones de variables. El escalar S incluye todas las variables discretas permitidas dispuestas en una secuencia ascendente. A cada elemento del escalar S se le da un número de secuencia que representa el valor de la variable discreta correspondiente. Esto se puede expresar así :

$$S_d = \{X_1, X_2, \dots, X_j, \dots, X_p\}, \quad 1 \leq j \leq p$$

Una función asignada $h(j)$ se selecciona el índice de los números de secuencia de los elementos del conjunto S y representa el valor X_j de las variables discretas en consecuencia.

$$h(j) = X_j$$

Así, los números de secuencia de los elementos serán un sustituto de los valores discretos en el escalar S . Este método se utiliza para buscar la solución óptima, y hace que las variables estén en un espacio de búsqueda continuo.

El PSO incluye un número de partículas, que son iniciadas aleatoriamente en el espacio de búsqueda. La posición de la partícula “i” puede ser descrita por el vector x_i .

$$x_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^D), \quad 1 \leq d \leq D, \quad i = 1, \dots, n$$

Donde D es la dimensión de la partícula, y n es la suma de todas las partículas. El escalar $x_i^d \in \{1, 2, \dots, j, \dots, p\}$ corresponde al grupo de variables discretas $\{X_1, X_2, \dots, X_j, \dots, X_p\}$ asignadas por la función $h(j)$. Por lo tanto, la partícula vuela a través de un espacio continuo, pero solo permanece en el espacio entero. En otras palabras todos los componentes del vector x_i son números enteros. Las posiciones de las partículas son actualizadas con base en la mejor posición personal de cada partícula, así como la mejor posición encontrada por el enjambre en cada iteración. La función objetivo es evaluada para cada partícula y el valor de fitness es usado para determinar qué posición en el espacio de búsqueda es mejor que las demás. El enjambre es actualizado por las siguientes ecuaciones:

$$V_i^{k+1} = wV_i^k + c_1r_1(P_i^k - x_i^k) + c_2r_2(P_g^k - x_i^k) \quad ()$$

$$x_i^{k+1} = INT(x_i^k + V_i^{k+1}) \quad ()$$

3.2.2.2 Parámetros del algoritmo PSO para la optimización del peso de la cercha.

- Numero de variables o dimensiones del problema “d”, esto equivale a en cuantos grupos se dividen los miembros de la cercha.
- Los limites del intervalo de búsqueda del enjambre de partículas, “inf” límite inferior y “sup” límite superior.

- El numero de partículas del enjambre “np”.
- Numero de datos para determinar la desviación estándar “rd”.
- Desviación estándar admisible “Est”.

Se definieron algunos parámetros estáticos dentro del algoritmo.

El factor de inercia “w”, este valor debe estar comprendido en un rango de valores de 0.9 y 1.2, con valores muy altos provoca que el algoritmo caiga en óptimos globales muy rápidamente y valores muy bajos el algoritmo se dirige a óptimos locales. Se escogió un valor inicial para la componente de inercia “ $w = 1$ ”, la cual irá disminuyendo hacia 0.98 en cada iteración del algoritmo.

Otros dos parámetros que se fijaron son las constantes de aprendizaje (“pesos”) de los componentes cognitivos y social de la velocidad de la partícula; “C1” es la constante de aprendizaje de la componente cognitiva de la velocidad de la partícula del enjambre y “C2” la constante de aprendizaje del componente social de la velocidad de la partícula, estudios realizados sobre la influencia de estas constantes de aprendizaje recomiendan que estos valores se fijen a “ $C1=C2=2$.” (J. Kennedy, Particle swarm optimization, December 1995).

Ya que el problema se compone por variables discretas, se hace un cambio de variable para el mejor funcionamiento del algoritmo. Los valores de cada posición de las partículas, desde la posición inicial hasta la final tomaran valores enteros comprendidos entre el límite inferior y el límite superior del intervalo de búsqueda. Estos valores equivalen a una coordenada de una matriz que guarda los valores de las propiedades de los perfiles utilizados en el diseño y en el cálculo de la función objetivo.

Definidos todos los parámetros, se generan las posiciones iniciales de las partículas, estas son distribuidas aleatoriamente en el espacio de búsqueda, definido por el usuario al inicio del algoritmo, posteriormente se calculan las velocidades iniciales de forma aleatoria para cada una de las partículas con base en el intervalo de búsqueda, y se define la máxima velocidad que puede tomar una partícula en cada movimiento. El intervalo está $[-V_{\max}, V_{\max}]$, donde el valor de la velocidad máxima se encuentra definido estáticamente como el rango de búsqueda dividido entre 10, no es recomendable fijar a cero la velocidad máxima ya que no se obtienen buenos resultados.

Una vez ya obtenida la posición inicial de cada partícula, se evalúan en la función objetivo y se verifica que las áreas obtenidas cumplan con las especificaciones de de diseño y así obtener un valor de del peso de la estructura. Se genera un vector "*Pm_local*", donde se almacena el peso mínimo de la cercha obtenido para cada partícula hasta el momento, en "*Pmin*" se guarda el peso mínimo obtenido hasta el momento, se crea el vector "*Pbest*" que almacena la posición que corresponden a las áreas que generan el peso mínimo hasta el momento.

Después de obtener cada una de las posiciones de la partícula donde obtuvo el menor valor del peso en el del enjambre de partículas, se actualizan las posiciones y velocidades de cada una, la velocidad nueva de la partícula es igual a la velocidad actual de la partícula multiplicada por el factor de inercia "w" más un componente cognitivo más un componente social. (Ver ecuación 28)

El componente cognitivo está asociado con el comportamiento de la partícula y está conformado por la resta de la mejor posición obtenida por la partícula con la posición actual de la partícula, esta resta se multiplicada por la constante de aprendizaje cognitivo y por un numero generado aleatoriamente entre 0 y 1. El componente social está asociado con el comportamiento del enjambre de partículas, si mi vecino es mejor mas tiendo a acercarme a mi vecino. Este factor

está conformado por la resta de la mejor posición obtenida por el cumulo de partículas con la posición actual de la partícula, esta resta se multiplicada por la constante de aprendizaje social y por un numero generado aleatoriamente entre 0 y 1. Después de actualizar la velocidad de la partícula se compara que la velocidad nueva no supere el límite máximo de velocidad establecido al inicio. Si la velocidad de la partícula es mayor a la velocidad máxima, la velocidad se iguala a la velocidad máxima permitida.

Una vez actualizada la velocidad de la partícula se procede a actualizar la posición, donde la nueva posición es igual a la posición actual más la velocidad de la partícula, el valor de la nueva posición se redondea ya que la posición de la partícula es una coordenada comprendida entre el límite inferior y el límite superior del espacio de búsqueda. El paso siguiente es verificar que la posición de la partícula no esté por fuera del intervalo de búsqueda, si la nueva posición se encuentra por fuera del intervalo de búsqueda, entonces la nueva posición de la partícula se igual a uno de los límites del intervalo de búsqueda donde este orientada la partícula.

El paso a seguir es evaluar las nuevas posiciones en la función objetivo, y se obtiene el peso de la estructura para cada partícula. Se determina el peso mínimo en cada iteración y se calcula la desviación estándar si la desviación estándar cumple con la establecida por el usuario al iniciar el algoritmo, este para y devuelve, el tiempo que dura el proceso, el número de iteraciones, el peso mínimo, las posiciones del mejor partícula (es decir las coordenadas), las áreas equivalentes a la posición de la mejor partícula y hace una grafica del “peso vs iteraciones”.

3.2.3 Descripción del algoritmo genético “GA” implementado en el programa Matlab® para la optimización del peso de la cercha.

Cuando se inicia el algoritmo genético “AG”, este algoritmo pide al usuario que ingrese algunos parámetros para su funcionamiento, estos parámetros son escogidos de acuerdo a la complejidad del problema que se quiere resolver.

Los parámetros a ingresar son:

- El tamaño de la población inicial “TamPob”.
- El numero de variables del problema “Var”.
- Los limites del intervalo de búsqueda, “Linf” límite inferior y “Lsup” límite superior.
- La probabilidad de cruce “Pc” y de mutación “Pm”.
- Número máximo de generaciones que se desee “Maxgen”.

Para la representación de cada individuo, se escogió la codificación binaria ya que esta facilita el uso de los operadores de cruce y mutación.

Para este caso las variables son discretas por lo que tenemos que manejar números enteros que están en un intervalo entre [1,32], a cada valor entero entre este intervalo está asociada un área de perfil comercial. Como tenemos 32 posibles valores en el intervalo entonces la longitud del digito binario es de 5, pero como manejaremos dos casos en los que las variables son 3 y 4 dependiendo de la agrupación de miembros que se estudie entonces la longitud total del individuo es de 15 y 20 para los dos casos estudiados de variables.

La tabla 4 muestra como se relacionan la cadena binaria con los valores discretos (Shyue-Jian Wu, 4 April 2000).

Tabla Relación entre la cadena binaria y los valores discretos

CADENA BINARIA	ENTEERO DECIMAL	PERFIL ASOCIADO
00000	0	1
00001	1	2
00010	2	3
00011	3	4
:	:	:
:	:	:
11110	30	31
11111	31	32

Fuente: Shyue-Jian Wu, P.-T. C. (4 April 2000). Integrated discrete and configuration optimization of trusses using genetic algorithms.

Figura Representación del cromosoma implementado en el programa

1	1	1	1	1	0	0	0	1	1	0	0	0	0	0
VARIABLE 1					VARIABLE 2					VARIABLE 3				

Fuente: Elaboración Propia

En la figura 11 podemos ver la representación de un individuo en codificación binaria, este está conformado por tres subcadenas de longitud 5, las cuales están asociadas a cada una de las variables del problema que estas a su vez están asociadas a un perfil de la lista de perfiles disponibles. Al igual que esta representación también tenemos una para cuatro variables de forma similar aumentando la cadena de 15 dígitos a 20.

El primer paso que realiza el algoritmo es generar de forma aleatoria la población inicial de individuos, que estén dentro del campo de búsqueda. Esto se hace agregando uno a uno números aleatorios entre 0 y 1, hasta completar la cadena total de representación para cada individuo en la población.

Se crea una estructura llamada “Población” y en unos de sus campos llamados “Pobvieja” se guarda la población inicial. El paso a seguir es decodificar el binario a número entero, estos valores son guardados en una matriz llamada “X0”. Los valores guardados en “X0” están asociados a un perfil en y le indica al programa que valores debe llevar cada elemento en cuanto a las áreas de la sección transversal.

Se evalúan los individuos en la función objetivo, es decir se evalúa el peso de la estructura, este peso se guarda en un vector llamado “Peso”, hay un contador llamado “Viol” que cuenta cuantas veces el individuo evaluado no cumple con las restricciones de diseño, este valor se guarda en un vector llamado “Violaciones”, esto me sirve para dar a cada individuo un valor de aptitud (fitness). El vector con nombre “Pminimo” va guardando la mejor solución, es decir el mejor individuo de cada generación.

La función de aptitud o de fitness que se escogida es la siguiente:

$$Fitness(i) = \frac{1}{Fobjetivo(i) * (Violaciones(i) + 1)}$$

Se puede ver que cuando el individuo no viola ninguna restricción entonces el fitness es igual al inverso del peso de la estructura, a mayor violaciones menor será la probabilidad de sobrevivir en la siguiente generación un individuo.

El siguiente paso es seleccionar los mejores individuos para formar la siguiente generación, a mayor aptitud mayor posibilidad de ser escogido para formar parte de la siguiente generación. El proceso de selección se hizo por medio de la ruleta.

Una vez escogidos los mejores individuos se procede a hacer el cruce entre ellos para esto se escogen dos padres y se les asigna un valor aleatorio entre 0 y 1 si este número es menor a la probabilidad de cruce " P_c " se procede a hacer el cruce entre ellos obteniendo dos individuos con las características de los padres, si el número aleatorio es mayor que " P_c " entonces pasan a la siguiente generación sin ser cruzados. El cruce se hace mediante el cruce uniforme.

Una vez hecho el paso anterior ahora se prosigue a mutar los individuos para esto se genera un número aleatorio entre 0 y 1, si este número está por debajo de la probabilidad de mutación " P_m " entonces este individuo pasa a ser mutado. La mutación se hace generando otro número aleatorio entre 0 y 1 para cada bit de la cadena del individuo si el número aleatorio es menor a " P_m " entonces el bit muta es decir si hay un 1 pasa a ser 0, o si hay un cero pasa a ser 1, si el número aleatorio es mayor a " P_m " entonces el bit no muta.

El mejor individuo de la generación pasa sin ser alterado esto se hace eliminando un individuo de la población nueva y agregando el mejor individuo de la generación anterior. La nueva población se guarda en el campo "Pobnueva" de la estructura "Poblacion".

Ya creada la nueva generación entonces la población nueva es evaluada en la función objetivo y se repite todo el proceso hasta cumplir con los criterios de parada en este caso hasta cumplir con el máximo número de generaciones propuesto por el usuario al iniciar el algoritmo.

El programa grafica al mejor individuo a lo largo de las generaciones.

4. RESULTADOS NUMÉRICOS

El presente capítulo presenta los resultados obtenidos por las modelaciones que se realizaron en el programa Matlab®, y se encuentran en el CD especificado con mayor claridad en el Anexo A.

4.1 RESULTADOS DE LA OPTIMIZACIÓN DEL PESO DE LA CERCHA EN ESTUDIO

Función objetivo: la función que utilizaremos para la optimización es el peso de la estructura

$$W(A) = \sum_{i=1}^n \rho A_i L_i$$

Donde $W(A)$ es el peso de la estructura en función de las áreas de los elementos, ρ es la densidad del material, A_i es el área del elemento i , L_i es la longitud del elemento i . En un diseño inicial propuesto por los autores, y haciendo una comparación de la fuerzas estáticas y dinámicas en la estructura se toma la decisión de despreciar las fuerzas producidas por un sismo ya que son insignificantes en comparación a las estáticas. Se hace un diseño inicial teniendo en cuenta la carga sobre los nodos de la cuerda superior de la cercha y el peso propio de la estructura multiplicada por un factor de seguridad de 1,2. Se verifico que cumpliera con los esfuerzos admisibles por la norma y los máximos desplazamientos admisibles. Este diseño inicial se hace para la agrupación de miembros del caso 1. En la tabla 5 se muestra el resultado de diseño.

Tabla Diseño inicial para agrupación de miembros del Caso 1

A1 [cm2]	A2 [cm2]	A3 [cm2]	PESO	Dmax
4,51	10,41	7,11	3866,43	-0,0143

Fuente: Elaboración propia

4.2 RESULTADOS DE LA OPTIMIZACIÓN MEDIANTE OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS

Una partícula es un vector compuesto por tres números enteros entre el límite inferior y el límite superior del espacio de búsqueda, en nuestro caso tenemos el intervalo [1,32], esto es la posición de la partícula en el espacio de búsqueda. Los datos en el vector de posición están asociados a una coordenada que representa el área de la sección transversal de las posibles soluciones. El enjambre es el grupo de partículas determinadas por el usuario para la búsqueda.

4.2.1 Resultados para el caso 1

Para la agrupación de miembros del caso 1, se hicieron pruebas para tres enjambres, de 2, 5, y 10 partículas, para los dos primeros hicimos 5 ejecuciones del algoritmo y para el ultimo hicimos tres ejecuciones. A continuación se muestran los resultados de las pruebas.

En la tabla 6 se muestra un resumen de los resultados obtenidos después de realizar las ejecuciones del algoritmo, se puede observar el tamaño del enjambre, el numero de iteraciones en cada prueba, las tres áreas iniciales escogidas aleatoriamente por el algoritmo, el desplazamiento máximo obtenido por la partícula en el inicio, el peso de la cercha al iniciar el algoritmo, luego tenemos las tres áreas finales es decir una vez terminada la optimización, en seguida se muestra el desplazamiento máximo de la cercha para el diseño optimo, luego encontramos el peso optimo de la estructura, y por ultimo tenemos el mejor peso para cada enjambre.

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el
Caso 1 (Análisis inicial)

PARTICULAS	ITERACIONES	AREA INICIAL [cm ²]			Dmax INICIAL	PESO INICIAL
		A ₁	A ₂	A ₃		
2	11	4,42	16,47	6,73	-0,0108	5309,03
	15	13,76	8,72	6,73	-0,0120	4721,98
	11	4,42	15,46	6,73	-0,0085	5049,20
	16	4,42	8,72	15,46	-0,0084	4222,28
	15	13,76	8,72	6,73	-0,0110	4721,98
5	6	4,42	8,72	6,73	-0,0157	3402,58
	9	5,49	8,72	6,73	-0,0117	3553,78
	11	6,73	8,72	6,73	-0,0126	3730,18
	11	4,42	8,72	10,41	-0,0094	3749,23
	9	4,42	8,72	8,03	-0,0125	3525,35
10	6	4,42	8,72	6,73	-0,0157	3402,58
	6	4,42	8,72	6,73	-0,0157	3402,58
	6	4,42	8,72	6,73	-0,0157	3402,58

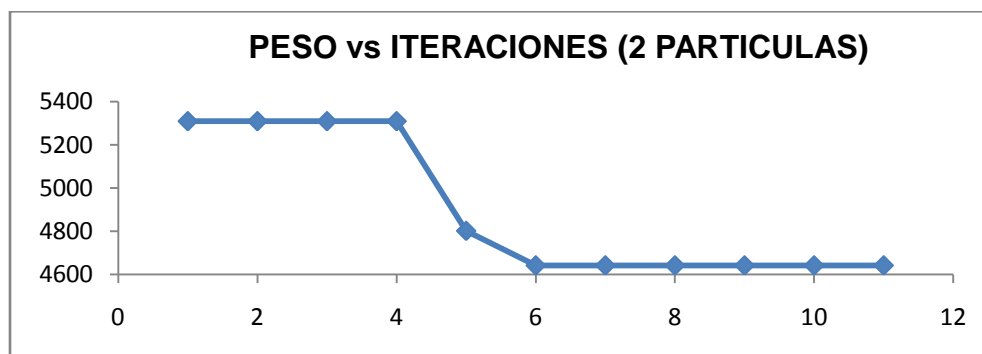
Fuente: Elaboración propia

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 1 (Análisis final y peso óptimo)

PARTICULAS	ITERACIONES	AREA FINAL [cm ²]			Dmax FINAL	PESO FINAL	PESO OPTIMO
		A ₁	A ₂	A ₃			
2	11	4,42	13,76	6,73	-0,0137	4642,24	3402,58
	15	4,42	8,72	6,73	-0,0157	3402,58	
	11	4,42	8,72	6,73	-0,0157	3402,58	
	16	4,42	8,72	6,73	-0,0157	3402,58	
	15	4,42	8,72	6,73	-0,0157	3402,58	
5	6	4,42	8,72	6,73	-0,0157	3402,58	3402,58
	9	4,42	8,72	6,73	-0,0157	3402,58	
	11	4,42	8,72	6,73	-0,0157	3402,58	
	11	4,42	8,72	6,73	-0,0157	3402,58	
	9	4,42	8,72	6,73	-0,0157	3402,58	
10	6	4,42	8,72	6,73	-0,0157	3402,58	3402,58
	6	4,42	8,72	6,73	-0,0157	3402,58	
	6	4,42	8,72	6,73	-0,0157	3402,58	

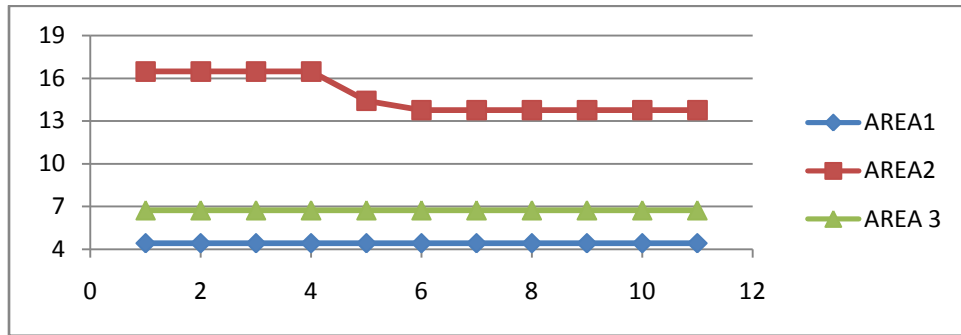
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



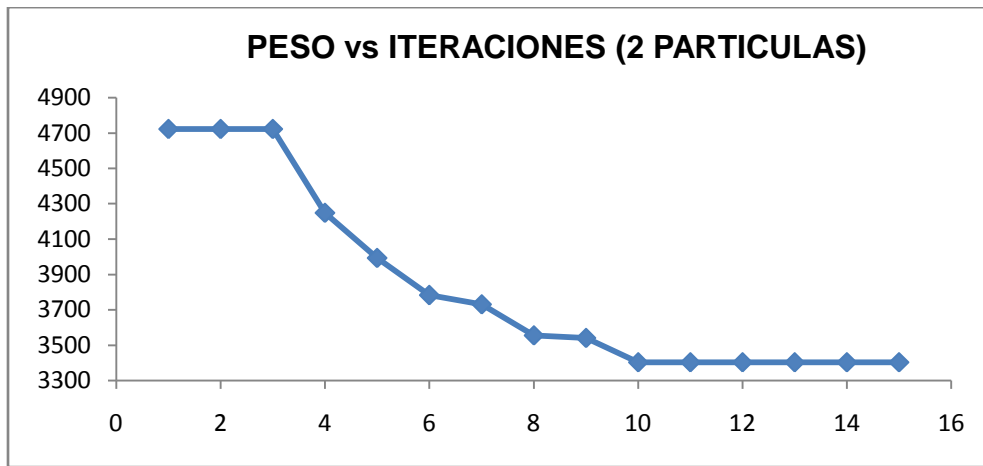
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



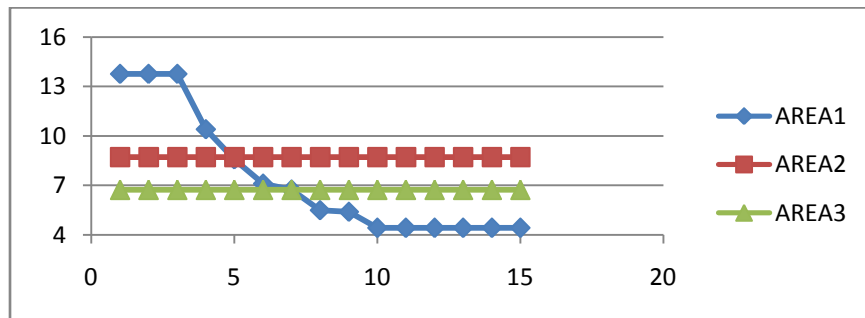
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



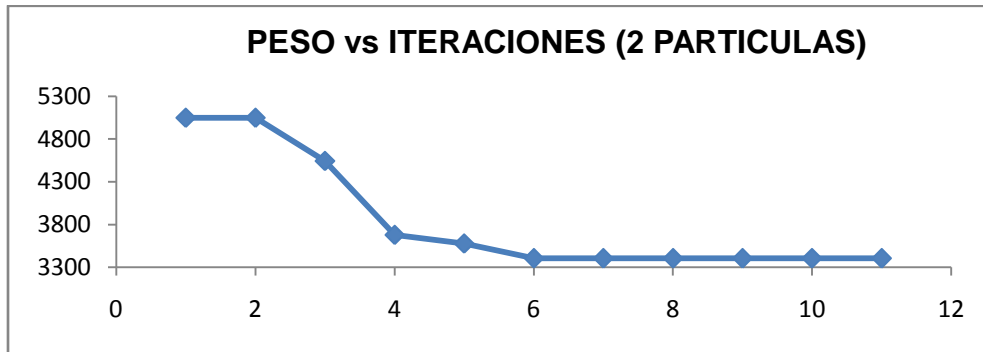
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



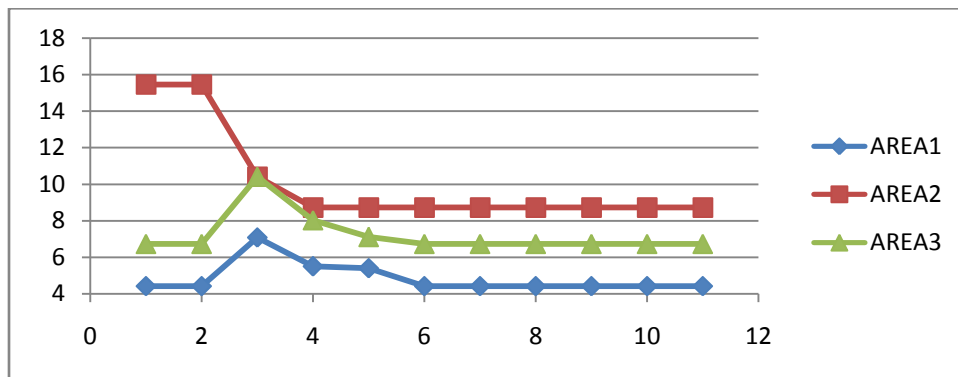
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



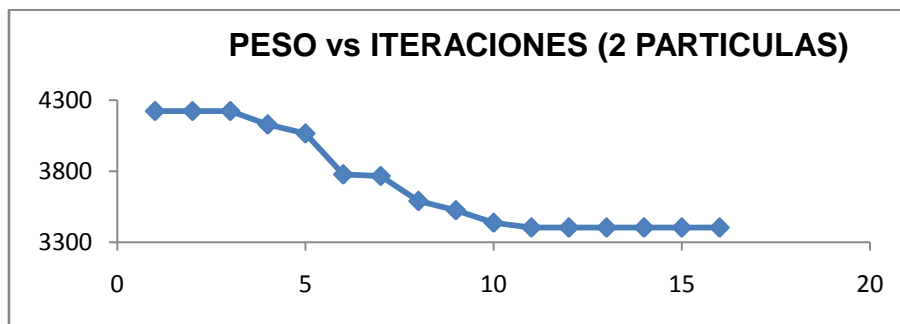
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



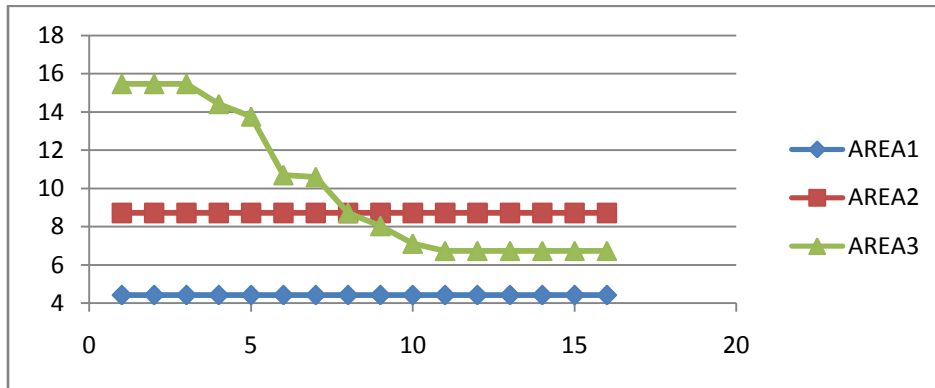
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



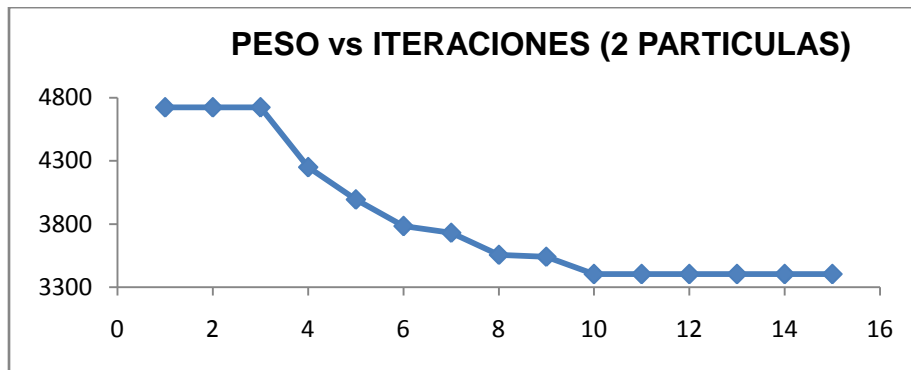
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



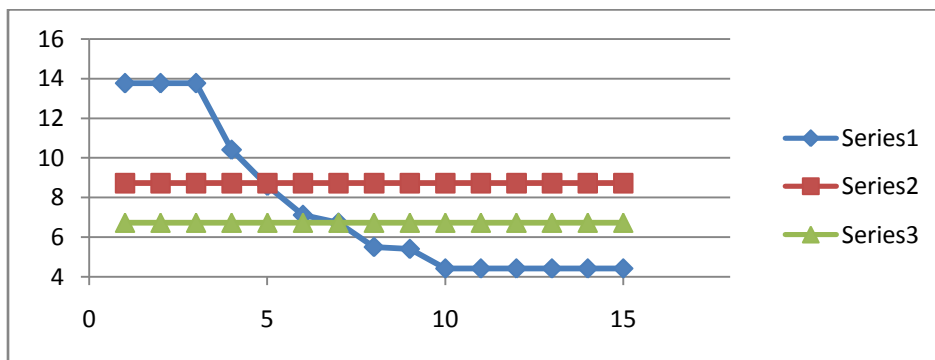
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 1)



Fuente: Elaboración propia

En la tabla 8 se muestra un resumen de los resultados óptimos para las cinco ejecuciones del algoritmo con un enjambre de dos partículas, además se muestra el tiempo que tardo el algoritmo en realizar la optimización.

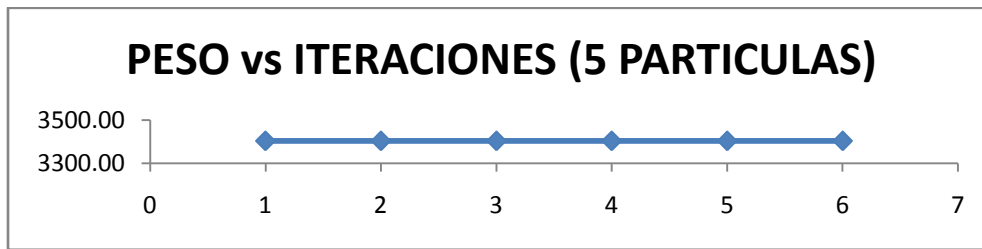
Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 2 partículas (Caso 1)

EJECUCION	TIEMPO [seg]	AREAS OPTIMAS			PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	
1	28	4,42	13,76	6,73	4642,24
2	31	4,42	8,72	6,73	3402,58
3	23	4,42	8,72	6,73	3402,58
4	31	4,42	8,72	6,73	3402,58
5	30	4,42	8,72	6,73	3402,58

Fuente: Elaboración propia

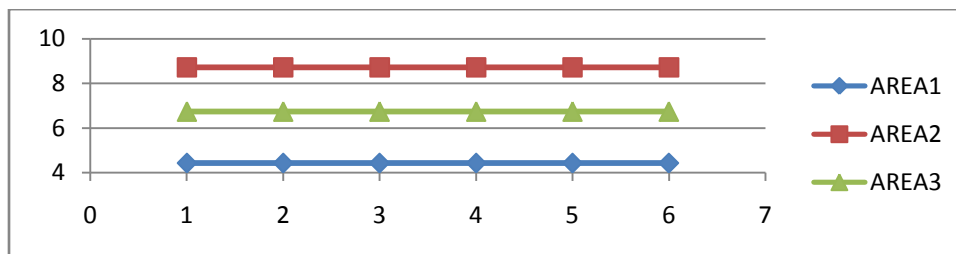
De acuerdo a los resultados obtenidos vemos que en la primera ejecución no se llega a un peso óptimo pero en las siguientes cuatro si se llega a un resultado optimo. Aun así podemos notar que el algoritmo en cada ejecución se comporta de manera diferente ya que en cuatro ejecuciones solo varia un área que no necesariamente siempre es la misma y en una ejecución vemos que varían las tres áreas. La ejecución del algoritmo para un enjambre de 2 partículas tiene un tiempo promedio de 29 segundos. Para este caso en el que el enjambre es de 2 partículas notamos que el peso óptimo se puede obtener entre la sexta y la undécima iteración.

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



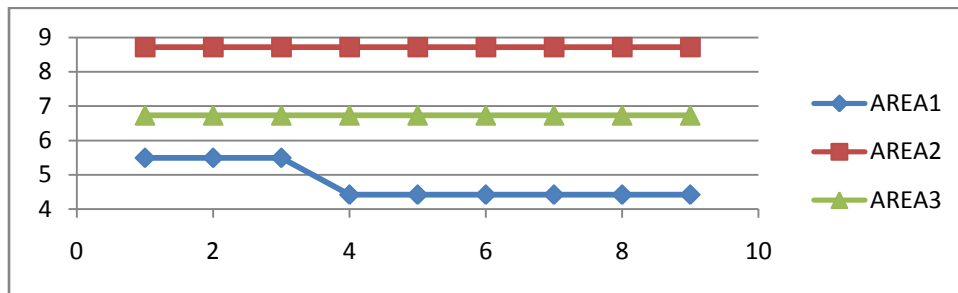
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



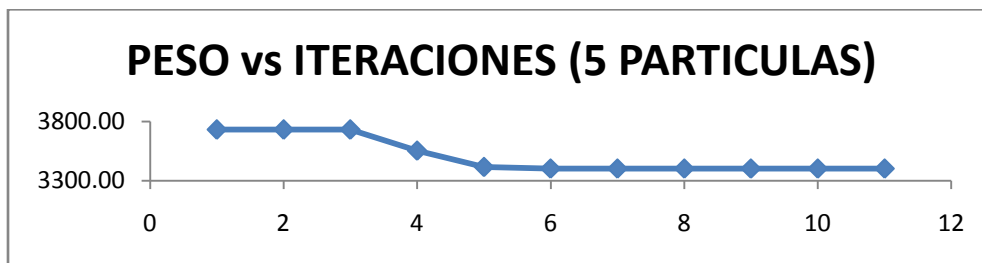
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



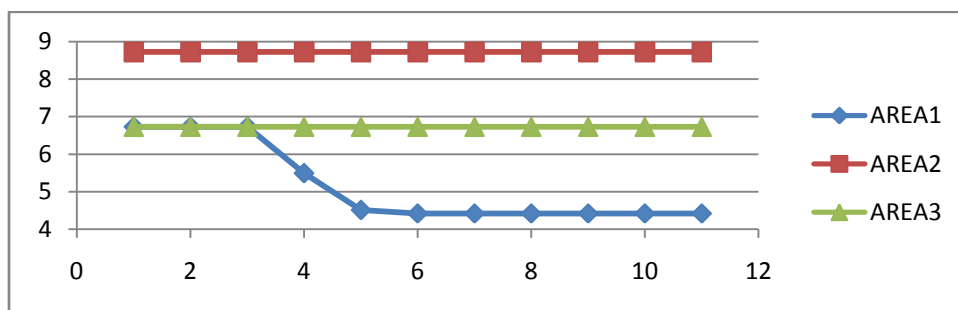
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



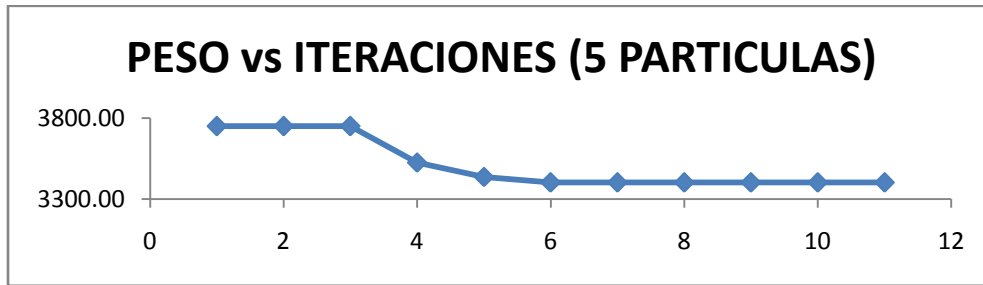
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



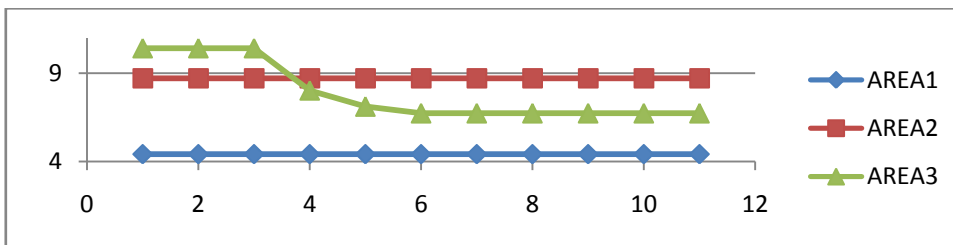
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



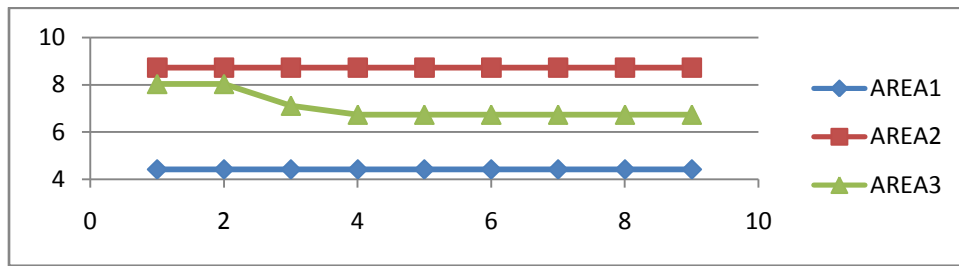
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 1)



Fuente: Elaboración propia

En la tabla 9 se muestra un resumen de los resultados óptimos para las cinco ejecuciones del algoritmo con un enjambre de cinco partículas, además se muestra el tiempo que tardó el algoritmo en realizar la optimización.

Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 5 partículas (Caso 1)

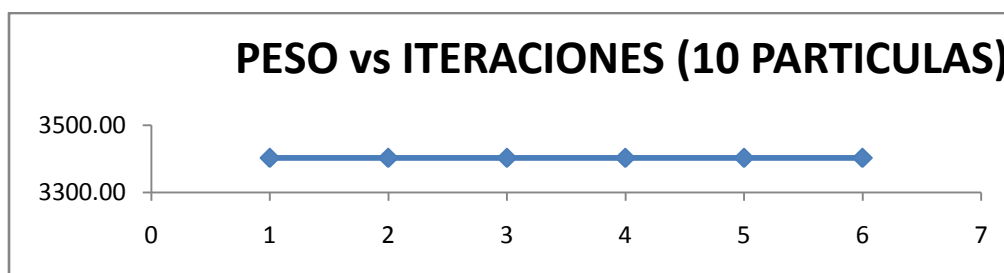
EJECUCION	TIEMPO [seg]	AREAS OPTIMAS			PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	
1	39	4,42	8,72	6,73	3402,58
2	51	4,42	8,72	6,73	3402,58
3	57	4,42	8,72	6,73	3402,58
4	53	4,42	8,72	6,73	3402,58
5	65	4,42	8,72	6,73	3402,58

Fuente: Elaboración propia

De acuerdo a los resultados obtenidos luego de ejecutar el algoritmo para un enjambre de 5 partículas, notamos que siempre llegamos al peso óptimo a diferencia de las ejecuciones con un enjambre dos partículas. Notamos que en la primera ejecución, el algoritmo en la primera iteración encuentra las áreas óptimas y por consiguiente el peso óptimo, esto se debe a que el algoritmo escoge aleatoriamente las primeras áreas para empezar la optimización. Se puede

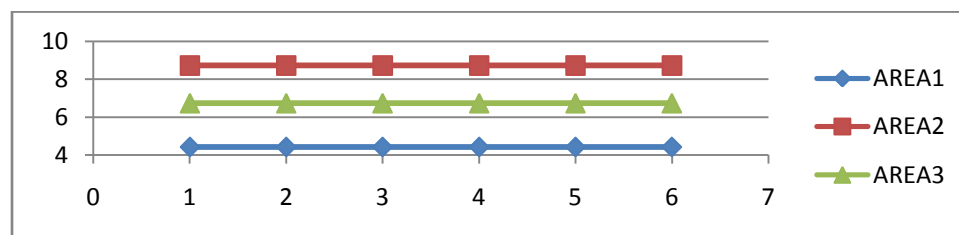
observar que de la ejecución 2 a la 5 solo varía un área durante la optimización pero que no siempre es la misma. El tiempo promedio de ejecución del algoritmo fue de 66 segundos. Para este enjambre de 5 partículas el peso óptimo se encontró entre la cuarta y la sexta iteración, aunque es posible obtenerlo en la primera iteración como ocurrió en la primera ejecución. A pesar de que para este enjambre se tiene un promedio de tiempo mayor notamos que el óptimo lo podemos obtener en menos iteraciones que para el enjambre de dos partículas.

Figura . Variación del peso en cada iteración para las tres ejecuciones del algoritmo con un enjambre de 10 partículas (Caso 1)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para las tres ejecuciones del algoritmo con un enjambre de 10 partículas (Caso 1)



Fuente: Elaboración propia

En la tabla 10 se muestra un resumen de los resultados óptimos para las tres ejecuciones del algoritmo con un enjambre de 10 partículas, además se muestra el tiempo que tardó el algoritmo en realizar la optimización.

Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 10 partículas (Caso 1)

EJECUCION	TIEMPO [seg]	AREAS OPTIMAS			PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	
1	74	4,42	8,72	6,73	3402,58
2	81	4,42	8,72	6,73	3402,58
3	85	4,42	8,72	6,73	3402,58

Fuente: Elaboración propia

Luego de obtener los resultados de las ejecuciones del algoritmo para un enjambre 10 partículas podemos notar que en las tres ejecuciones el algoritmo encuentra el óptimo en la primera iteración, esto se debe a que el espacio de búsqueda se hace mucho más grande en comparación a los otros dos enjambres.

En la tabla 11 se muestra la comparación del diseño inicial propuesto por los autores y el diseño optimo obtenido después de ejecutar el algoritmo para los diferentes enjambres.

Tabla Comparación del diseño inicial y el diseño óptimo obtenido con el PSO. (Caso 1)

DISEÑO	A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	Dmax	PESO	% REDUCCION DEL PESO
INICIAL	4,51	10,41	7,11	-0,0143	3866,43	
OPTIMO	4,42	8,72	6,73	-0,0157	3402,58	12,00

Fuente: Elaboración propia

Podemos ver que si se reduce el peso de la estructura comparado con el diseño inicial propuesto por los autores, se disminuyen las tres áreas y aunque el desplazamiento máximo aumenta, este no sobrepasa el límite de 20 mm.

Para todos los enjambres obtuvimos los mismos resultados del peso óptimo a pesar que una ejecución del algoritmo con dos partículas no dio el resultado esperado.

4.2.2 Resultados para el caso 2

Para la agrupación de miembros del caso 2, se hicieron pruebas para 5 enjambres, de 2, 5, 10, 15, 30, y 50 partículas cada uno, cinco ejecuciones del algoritmo para los tres primeros enjambres, dos ejecuciones para los enjambres de 15 y 30 partículas y una ejecución para el enjambre de 50 partículas. A continuación se muestran los resultados de las pruebas.

En la tabla 12 se muestra un resumen de los resultados obtenidos después de realizar las ejecuciones del algoritmo, se puede observar el tamaño del enjambre, el número de iteraciones en cada prueba, las cuatro áreas iniciales escogidas aleatoriamente por el algoritmo, el desplazamiento máximo obtenido por la partícula en el inicio, el peso de la cercha al iniciar el algoritmo, luego tenemos las cuatro áreas finales es decir una vez terminada la optimización, en seguida se muestra el desplazamiento máximo de la cercha para el diseño óptimo, luego encontramos el peso óptimo de la estructura, y por último tenemos el mejor peso para cada enjambre.

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 2 (Análisis inicial)

PARTICULAS	ITERACIONES	AREA INICIAL [cm ²]				Dmax INICIAL	PESO INICIAL
		A ₁	A ₂	A ₃	A ₄		
2	16	4,51	4,42	6,73	7,11	-0,0152	2384,82
	25	16,47	4,42	13,76	7,11	-0,0099	4738,24
	19	5,49	4,42	14,41	7,11	-0,0144	3248,03
	19	4,51	4,42	15,46	13,76	-0,0091	3265,39
	17	6,54	4,42	10,69	7,11	-0,0111	3044,16
5	21	7,11	5,49	6,73	10,59	-0,0154	3037,02
	20	4,42	4,42	14,41	8,72	-0,0094	3111,52
	15	4,42	5,49	6,73	10,59	-0,0152	2657,22
	14	4,42	4,42	8,72	7,11	-0,0152	2558,79
	14	4,42	4,42	7,11	7,11	-0,0144	2407,13
10	18	5,5	5,5	8,72	7,11	-0,0135	2967,96
	18	4,51	5,49	6,73	8,72	-0,0155	2652,68
	14	4,42	4,42	7,11	7,11	-0,0156	2407,13
	17	4,42	5,49	6,73	10,59	-0,0123	2657,22
	15	4,42	5,49	6,73	7,11	-0,0146	2625,38
15	16	4,42	5,49	6,73	7,11	-0,0156	2625,38
	11	4,42	4,42	6,73	7,11	-0,0157	2372,22
30	15	5,49	5,49	8,03	7,11	-0,0133	2899,36
	14	4,42	4,42	7,11	7,11	-0,0159	2407,13
50	11	4,42	4,42	6,73	7,11	-0,0156	2372,22

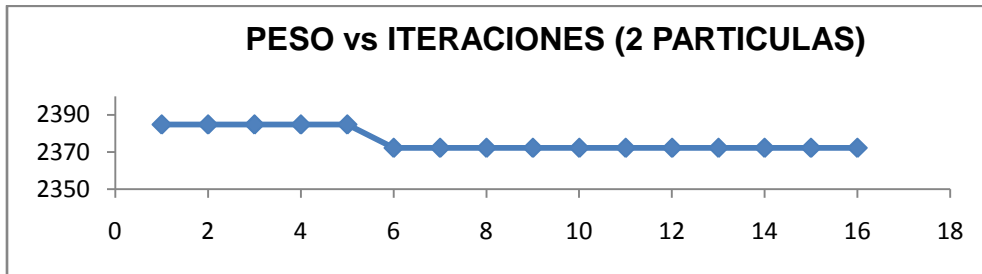
Fuente: Elaboración propia

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo PSO para el Caso 2 (Análisis final y peso óptimo)

PARTICULAS	ITERACIONES	AREA FINAL [cm ²]				Dmax FINAL	PESO FINAL	PESO OPTIMO
		A ₁	A ₂	A ₃	A ₄			
2	16	4,42	4,42	6,73	7,11	-0,0184	2372,22	2372,22
	25	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	19	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	19	4,42	4,42	6,73	8,72	-0,0178	2386,91	
	17	4,42	4,42	6,73	7,11	-0,0184	2372,22	
5	21	4,42	4,42	6,73	8,72	-0,0178	2386,91	2372,22
	20	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	15	4,42	4,42	6,73	8,72	-0,0178	2386,91	
	14	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	14	4,42	4,42	6,73	7,11	-0,0184	2372,22	
10	18	4,42	4,42	6,73	7,11	-0,0184	2372,22	2372,22
	18	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	14	4,42	4,42	6,73	7,11	-0,0184	2372,22	
	17	4,42	4,42	6,73	8,72	-0,0178	2386,91	
	15	4,42	4,42	6,73	7,11	-0,0184	2372,22	
15	16	4,42	4,42	6,73	7,11	-0,0184	2372,22	2372,22
	11	4,42	4,42	6,73	7,11	-0,0184	2372,22	
30	15	4,42	4,42	6,73	7,11	-0,0184	2372,22	2372,22
	14	4,42	4,42	6,73	7,11	-0,0184	2372,22	
50	11	4,42	4,42	6,73	7,11	-0,0156	2372,22	2372,22

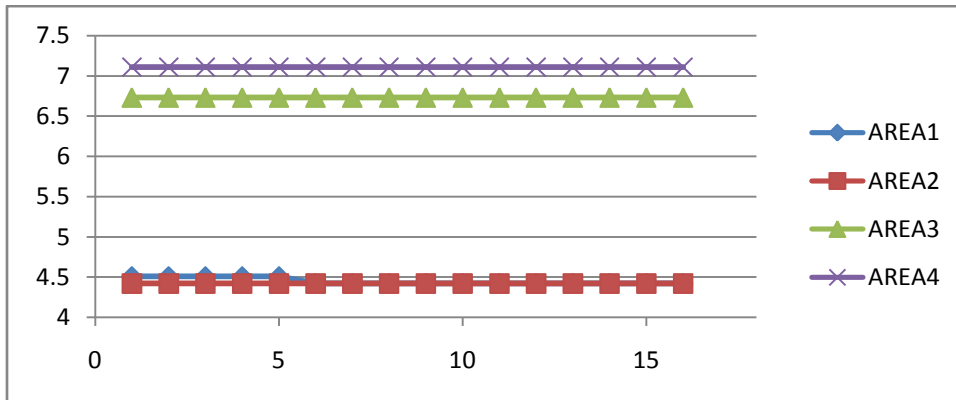
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



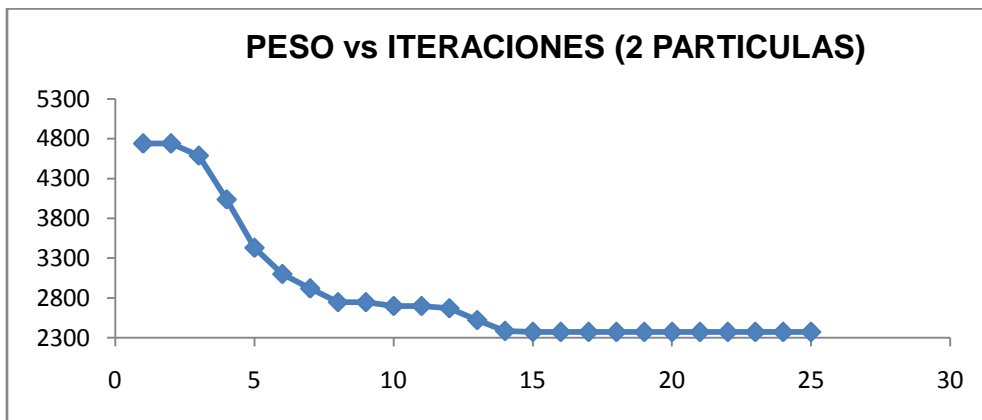
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



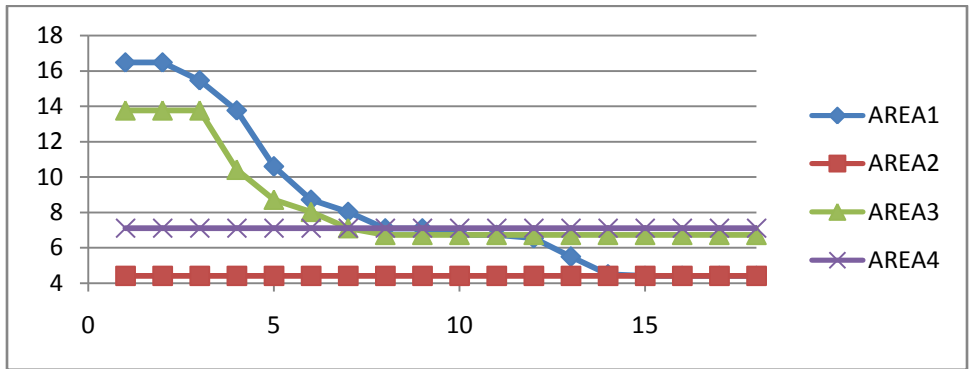
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



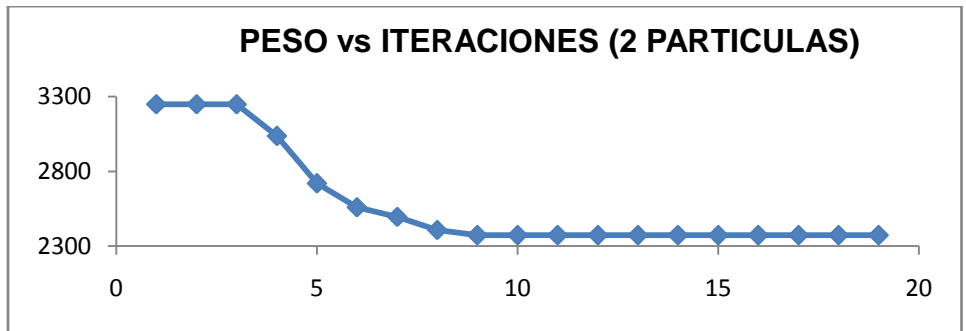
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



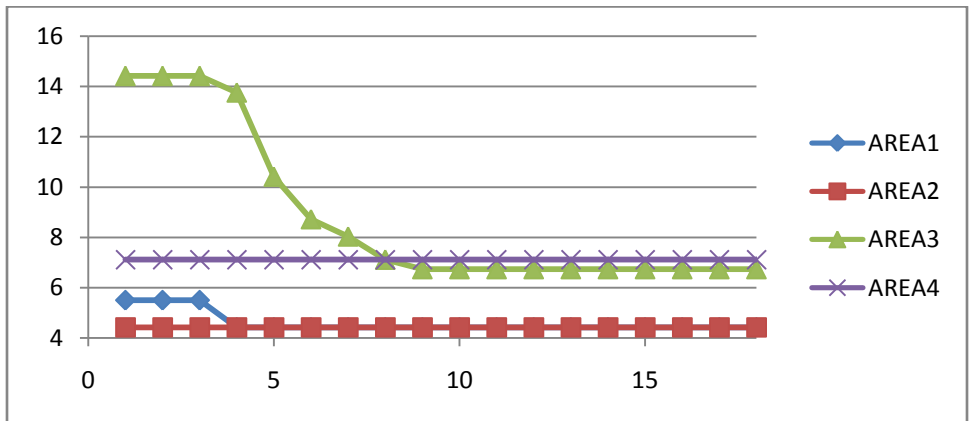
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



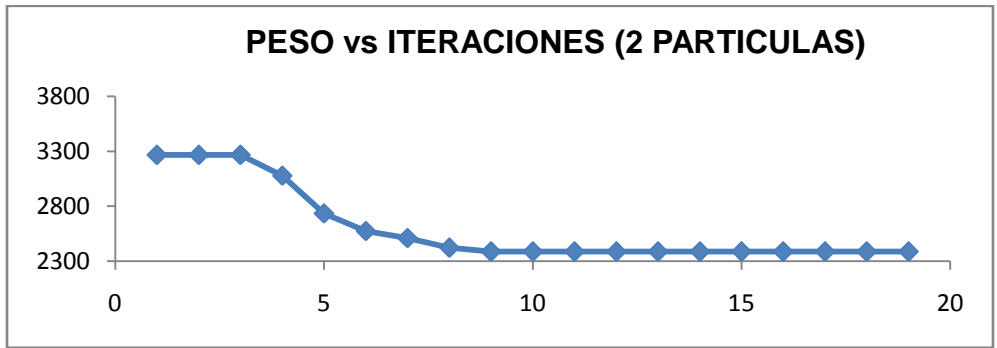
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



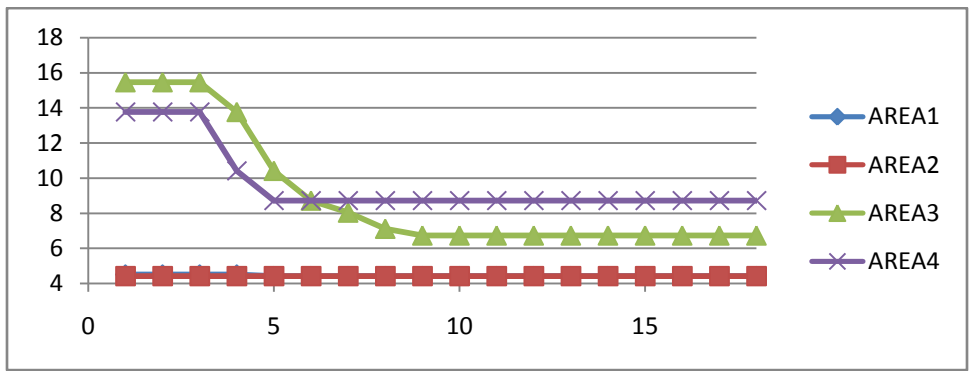
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



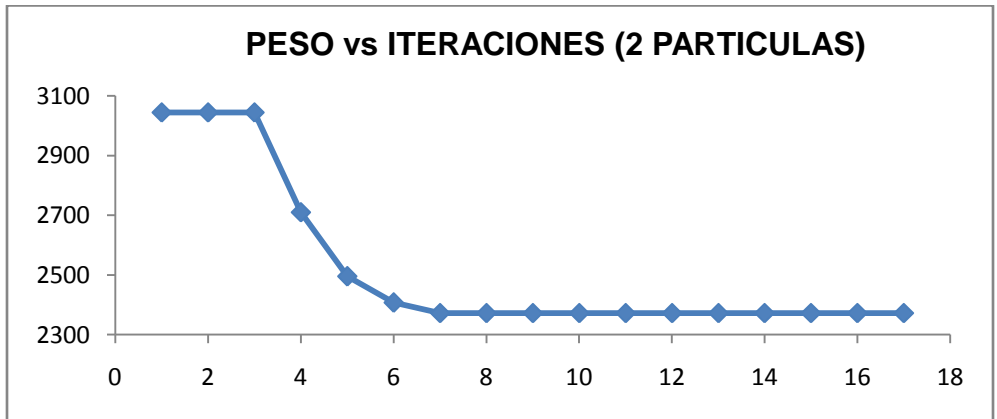
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



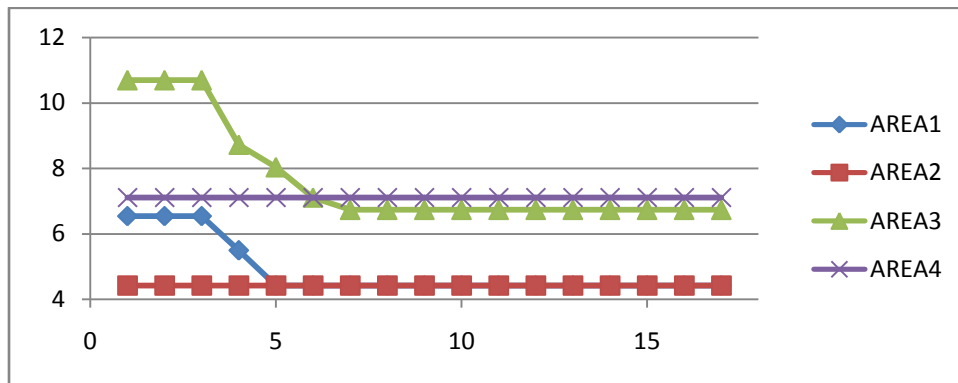
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 2 partículas (Caso 2)



Fuente: Elaboración propia

En la tabla 14 se muestra un resumen de los resultados óptimos para las cinco ejecuciones del algoritmo con un enjambre de dos partículas, además se muestra el tiempo que tardo el algoritmo en realizar la optimización.

Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 2 partículas (Caso 2)

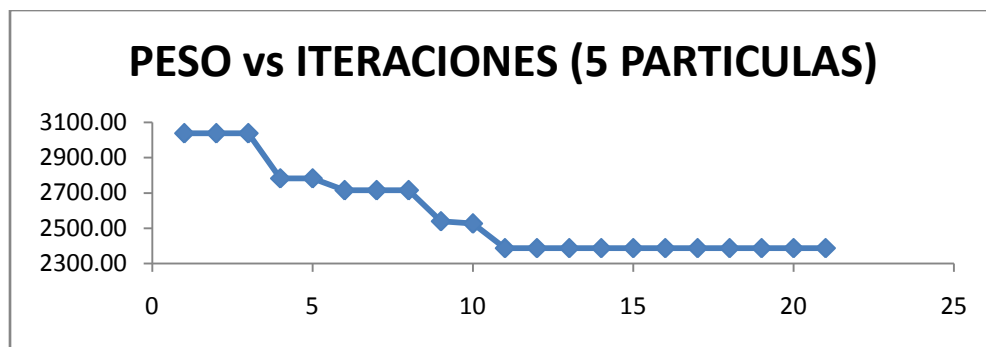
EJECUCION	TIEMPO [seg]	AREAS OPTIMAS				PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	A4 [cm ²]	
1	48	4,42	4,42	6,73	7,11	2372,22
2	27	4,42	4,42	6,73	7,11	2372,22
3	29	4,42	4,42	6,73	7,11	2372,22
4	25	4,42	4,42	6,73	8,72	2386,91
5	32	4,42	4,42	6,73	7,11	2372,22

Fuente: Elaboración propia

De acuerdo a los resultados obtenidos podemos ver que el algoritmo hace una buena optimización para un enjambre de dos partículas, a pesar de que en la cuarta ejecución no da un valor óptimo, pero se encuentra muy cercano. Podemos

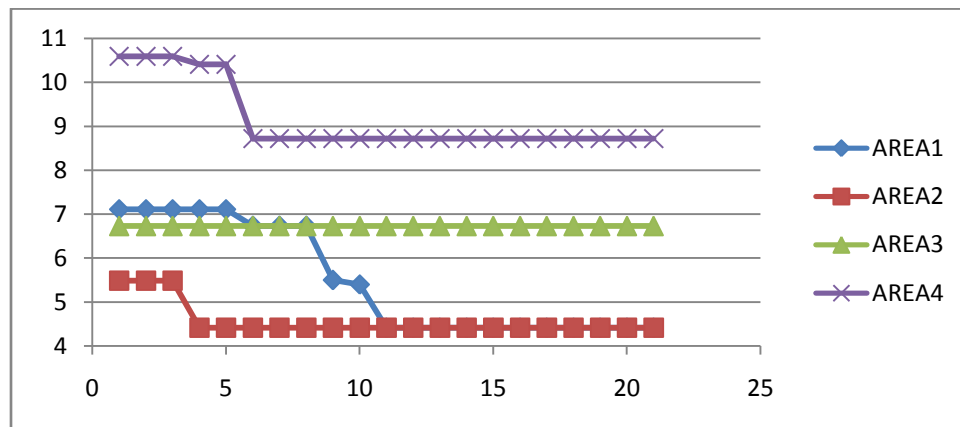
observar que para todas las ejecuciones el área 2 siempre obtuvo el óptimo en la primera ejecución, las demás áreas variaron dependiendo de la ejecución aunque en ocasiones también obtenían la óptima en la primera iteración. El tiempo promedio de ejecución del algoritmo fue de 32 segundos. Para este enjambre de 2 partículas el peso óptimo se encontró entre la sexta y la decimo quinta iteración.

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



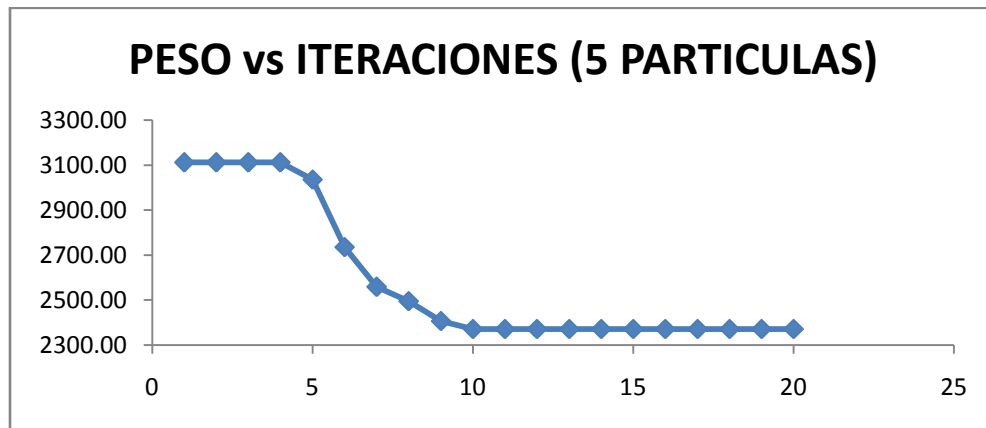
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



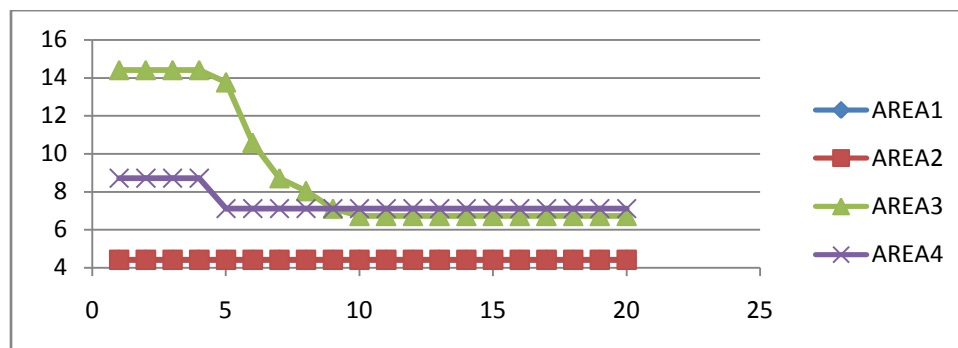
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



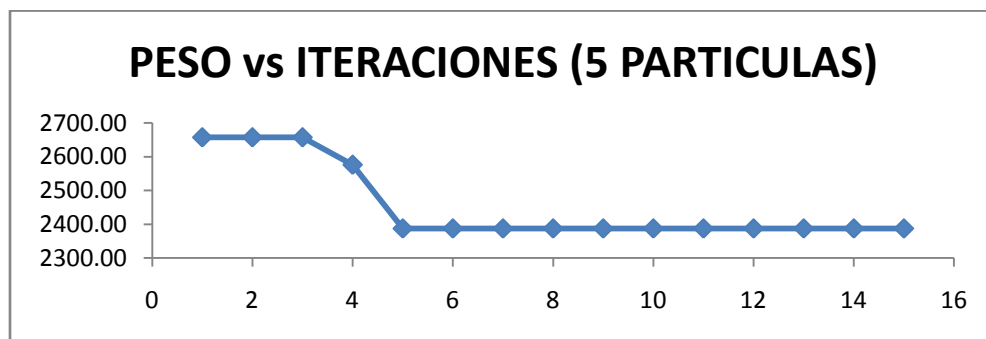
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



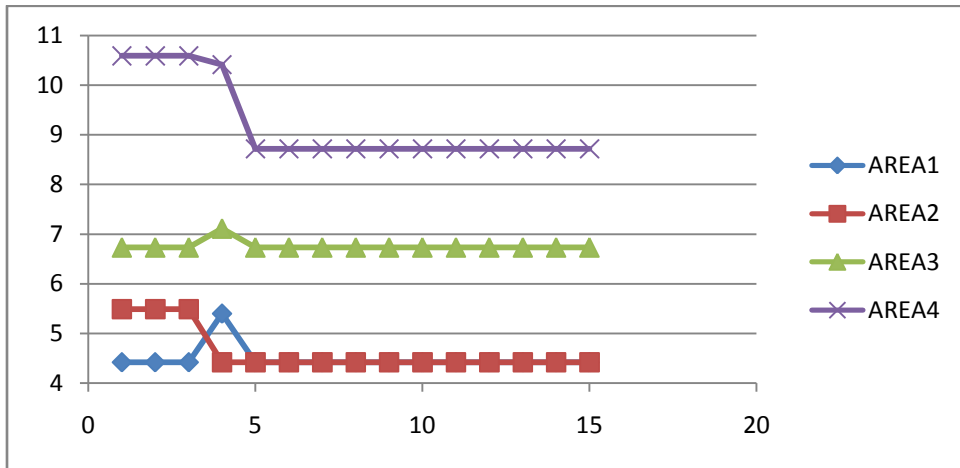
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



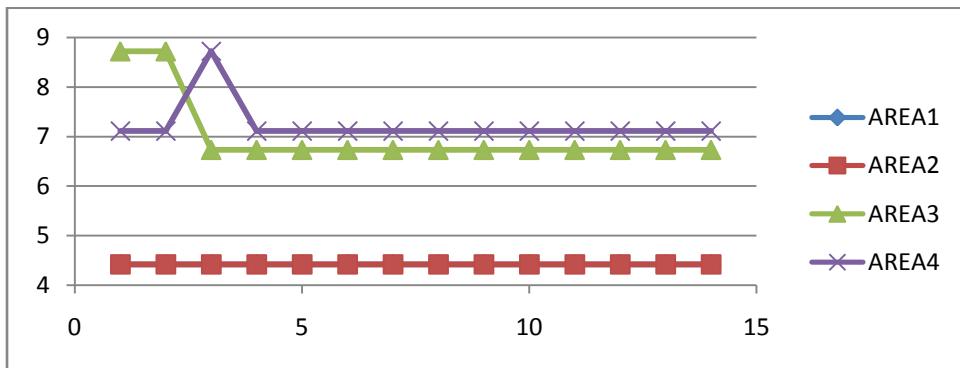
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



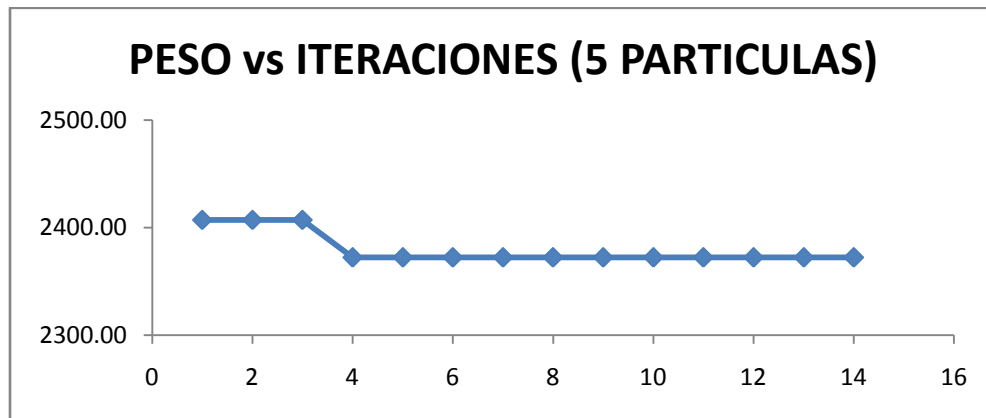
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



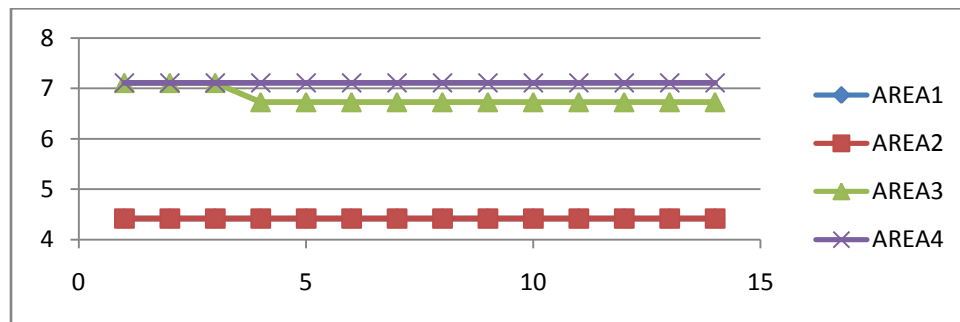
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 5 partículas (Caso 2)



Fuente: Elaboración propia

En la tabla 15 se muestra un resumen de los resultados óptimos para las cinco ejecuciones del algoritmo con un enjambre de cinco partículas, además se muestra el tiempo que tardo el algoritmo en realizar la optimización.

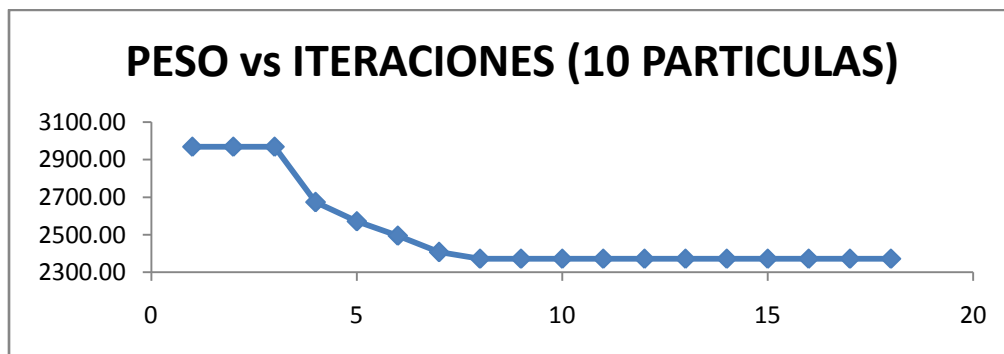
Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 5 partículas (Caso 2)

EJECUCION	TIEMPO [seg]	AREAS OPTIMAS				PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	A4 [cm ²]	
1	74	4,42	4,42	6,73	8,72	2386,91
2	71	4,42	4,42	6,73	7,11	2372,22
3	74	4,42	4,42	6,73	8,72	2386,91
4	71	4,42	4,42	6,73	7,11	2372,22
5	87	4,42	4,42	6,73	7,11	2372,22

Fuente: Elaboración propia

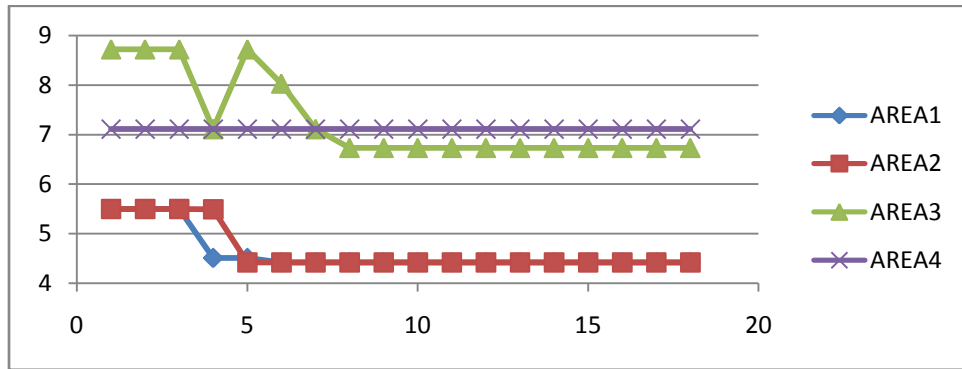
Podemos observar que luego de las cinco ejecuciones para un enjambre de 5 partículas, el algoritmo hace de nuevo la optimización en tres ocasiones y pero en dos ejecuciones el peso mínimo obtenido es muy cercano al óptimo. En esta ocasión la variación de las áreas fue total, es decir variaban todas dependiendo de la ejecución del algoritmo no hubo un comportamiento constante en las cinco ejecuciones. El tiempo promedio de la ejecución del algoritmo fue de 76 segundos el doble del tiempo promedio para un enjambre de dos partículas. El peso óptimo se pudo encontrar entre la cuarta y la quinta iteración. Podemos observar que aunque tiene una mayor duración el algoritmo encuentra el óptimo en menos iteraciones que en un enjambre de dos partículas.

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



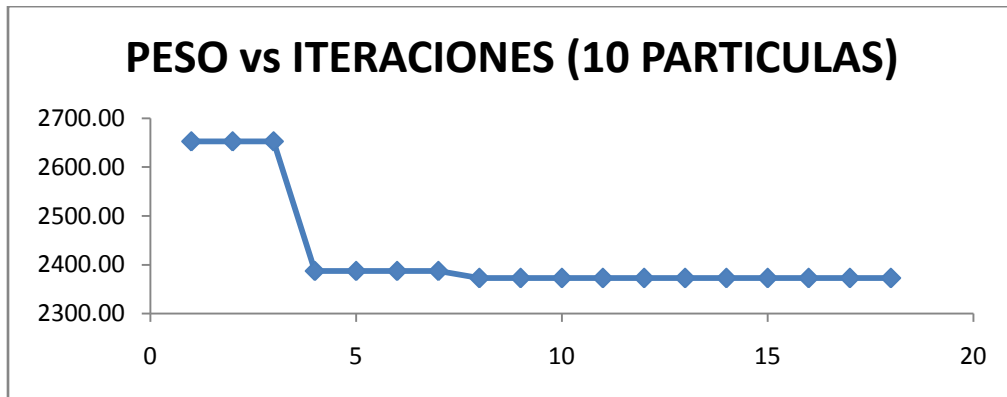
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



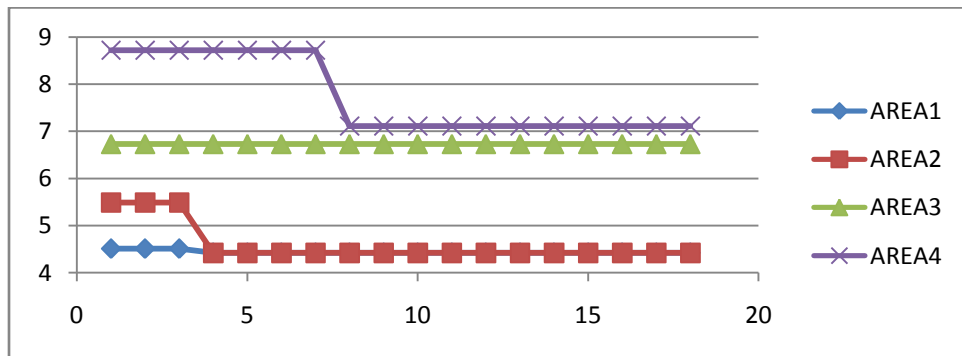
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



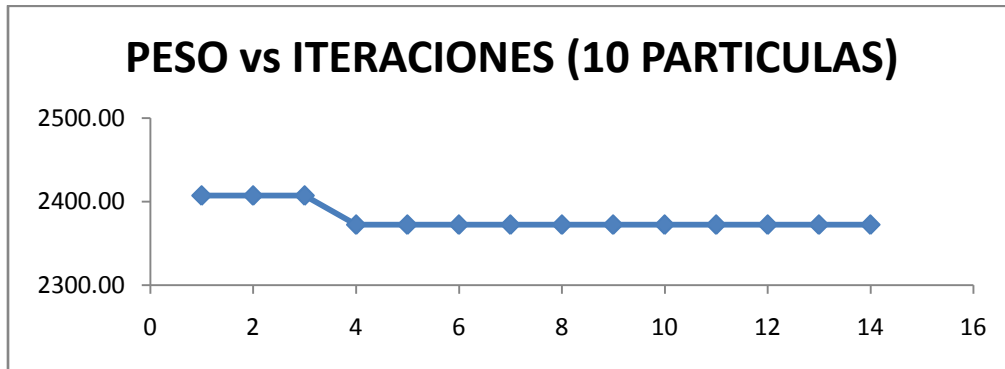
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



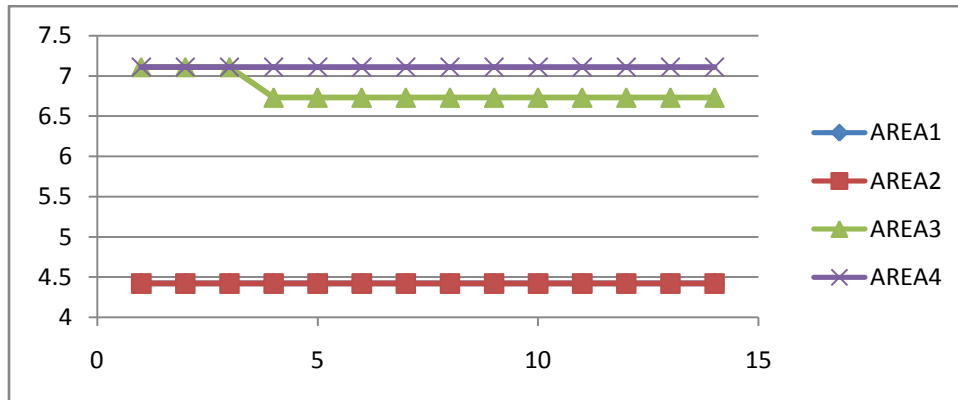
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la tercera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



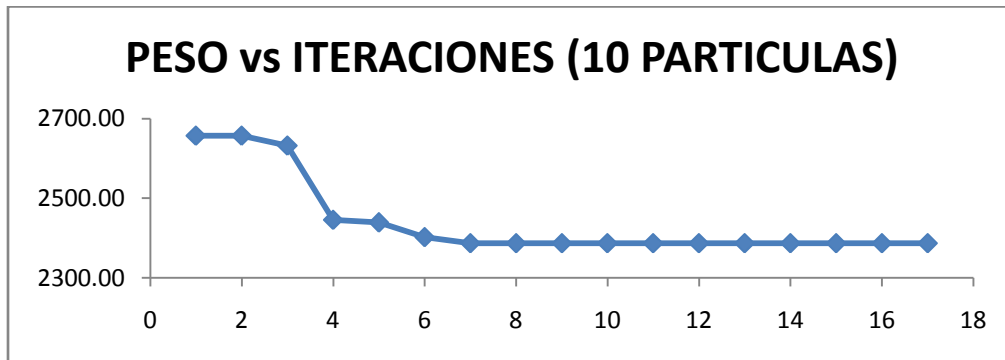
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la tercera ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



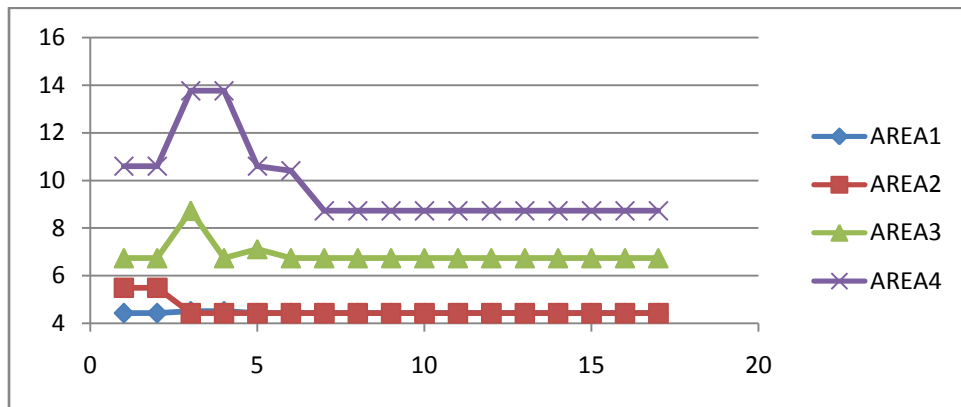
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



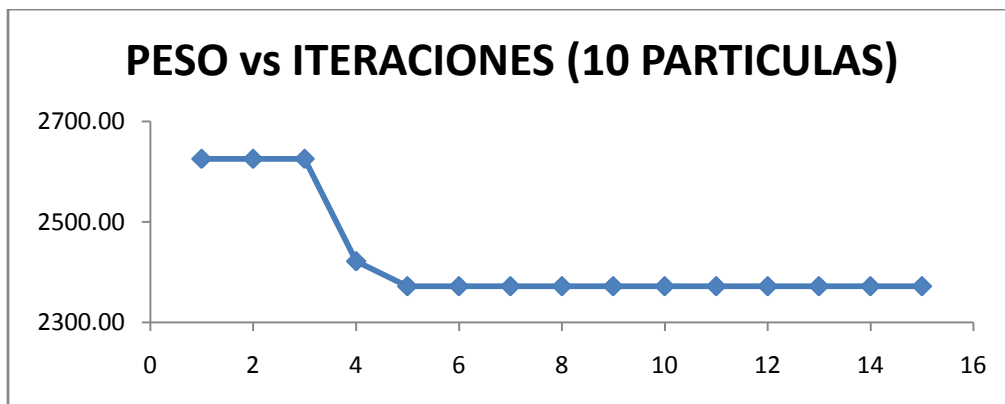
Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la cuarta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



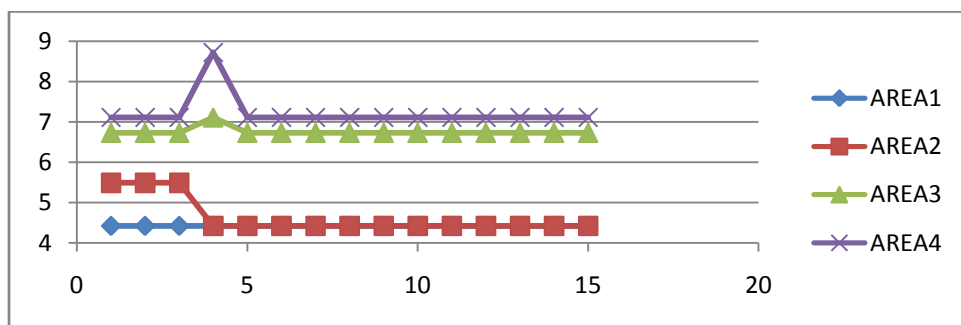
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la quinta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la quinta ejecución del algoritmo con un enjambre de 10 partículas (Caso 2)



Fuente: Elaboración propia

En la tabla 16 se muestra un resumen de los resultados óptimos para las cinco ejecuciones del algoritmo con un enjambre de diez partículas, además se muestra el tiempo que tardo el algoritmo en realizar la optimización.

Tabla Áreas y pesos óptimos obtenidos en cada ejecución del algoritmo para un enjambre de 10 partículas (Caso 2)

EJECUCION	TIEMPO [seg]	AREAS OPTIMAS				PESO OPTIMO
		A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	A4 [cm ²]	
1	141	4,42	4,42	6,73	7,11	2372,22
2	142	4,42	4,42	6,73	7,11	2372,22
3	139	4,42	4,42	6,73	7,11	2372,22
4	139	4,42	4,42	6,73	8,72	2386,91
5	135	4,42	4,42	6,73	7,11	2372,22

Fuente: Elaboración propia

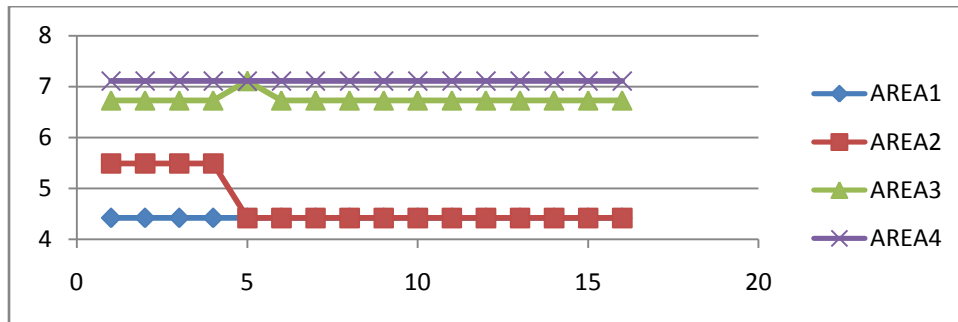
Al igual que para los dos enjambres anteriores hubo al menos una ejecución en la que el peso mínimo obtenido no es el optimo pero si muy cercano, en las restantes ejecuciones se obtiene el peso optimo. Vemos que el comportamiento de las áreas varía de ejecución en ejecución. El tiempo promedio de la ejecución del algoritmo fue de 136 segundos el doble del tiempo promedio para un enjambre de cinco partículas. El peso óptimo se pudo encontrar entre la cuarta y la octava iteración. El tiempo de duración es el doble pero podemos encontrar el peso óptimo en menos iteraciones que en el enjambre de cinco partículas.

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)



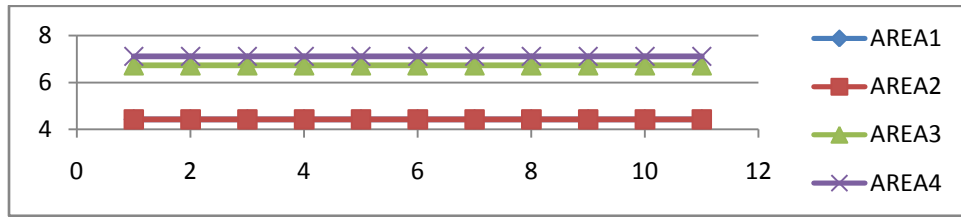
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)



Fuente: Elaboración propia

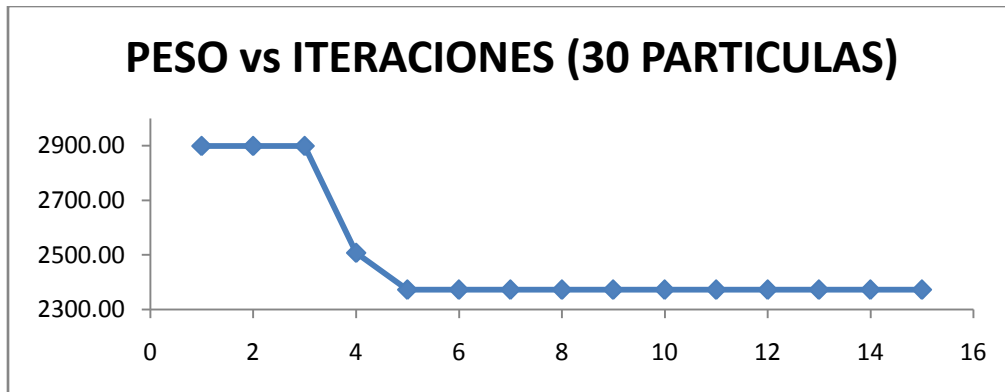
Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 15 partículas (Caso 2)



Fuente: Elaboración propia

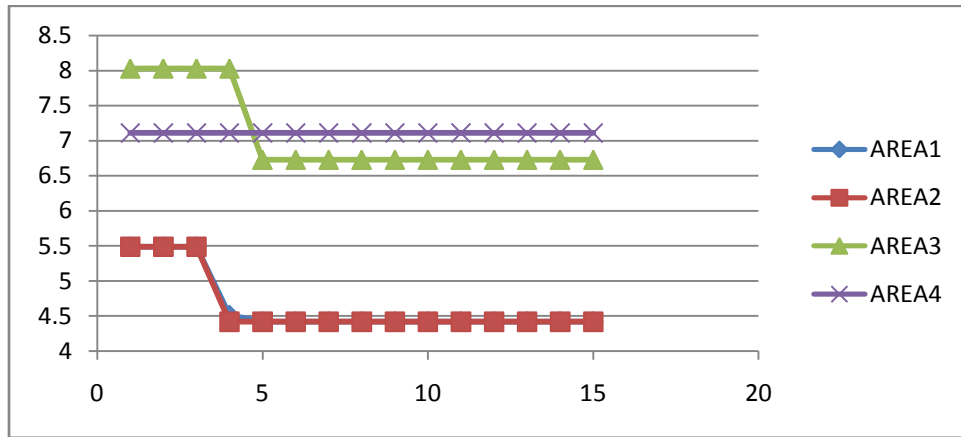
Obtenidos los resultados para las dos ejecuciones con un enjambre de 15 partículas, podemos ver que las dos iteraciones llegan al peso óptimo. En ambas ejecuciones las áreas 1 y 4 encuentran el óptimo desde la primera iteración. El tiempo promedio de ejecución del algoritmo es de 220 segundos es mucho más tiempo que en los anteriores enjambres pero podemos notar que el óptimo se puede lograr en la primera iteración.

Figura . Variación del peso en cada iteración para la primera ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la primera ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)



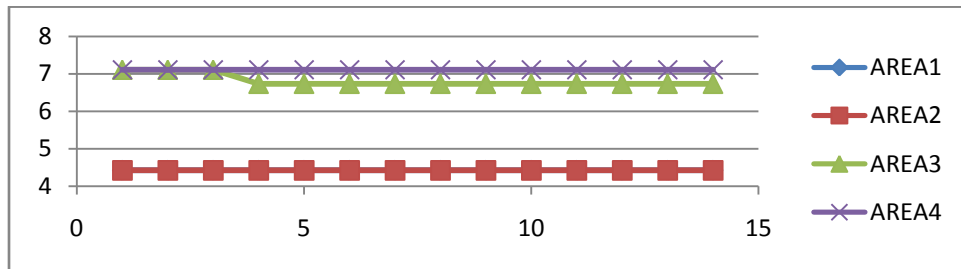
Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la segunda ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)



Fuente: Elaboración propia

Figura . Variación de las áreas en cada iteración para la segunda ejecución del algoritmo con un enjambre de 30 partículas (Caso 2)

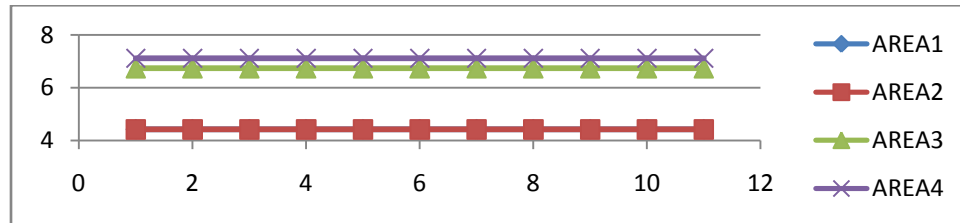


Fuente: Elaboración propia

Figura . Variación del peso en cada iteración para la ejecución del algoritmo con un enjambre de 50 partículas (Caso 2)



Figura . Variación de las áreas en cada iteración para la ejecución del algoritmo con un enjambre de 50 partículas (Caso 2)



Fuente: Elaboración propia

En la tabla 17 se muestra el diseño óptimo para el caso 2.

Tabla Diseño óptimo obtenido por el PSO para el Caso 2

AREAS OPTIMAS				PESO OPTIMO
A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	A4 [cm ²]	
4,42	4,42	6,73	7,11	2372,21864

En la tabla 18 se muestra el porcentaje de reducción del peso entre el caso 1 y caso 2 de la agrupación de miembros.

Tabla Porcentaje de reducción del peso entre los dos casos de agrupación de miembros

DISEÑO OPTIMO	PESO	%
CASO 1	3402,58	
CASO 2	2372,22	30,28

Es considerable la reducción del peso aumentando a 4 grupos las variables de diseño.

4.3 RESULTADOS DE LA OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS

Al igual que para la optimización con el PSO, con el algoritmo genético también evaluamos los dos casos de agrupación miembros explicados anteriormente. A continuación los resultados.

4.3.1 Resultados para el caso 1

Para la agrupación de miembros del caso 1, se hicieron pruebas para cinco poblaciones de 10, 20, 40, 60 y 100 individuos cada una, para todas las poblaciones se hicieron pruebas con 10, 20, 40, 60 y 100 generaciones.

En la tabla 19 se muestra un resumen de los resultados obtenidos después de realizar las ejecuciones del algoritmo, se puede observar el tamaño de la población, las generaciones en cada prueba, las tres áreas iniciales escogidas aleatoriamente por el algoritmo, el peso de la cercha al iniciar el algoritmo, luego tenemos las tres áreas finales es decir una vez terminada la optimización, luego encontramos el peso óptimo de la estructura, y por último tenemos el mejor peso para cada población.

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 1(Análisis inicial)

POBLACION	GENERACIONES	TIEMPO [Seg]	AREA INICIAL [cm ²]			PESO INICIAL
			A ₁	A ₂	A ₃	
10	10	15	7,08	16,47	16,47	6604,83
	20	30	7,11	20,41	33,63	9200,22
	40	60	20,41	20,41	6,73	8540,27
	60	79	10,59	7,08	14,41	4597,69
	100	106	10,69	8,03	6,73	4122,26
20	10	24	10,69	14,41	15,46	6507,19
	20	46	20,41	15,46	8,72	7496,57
	40	93	25,65	13,76	6,73	7642,84
	60	137	5,49	20,41	14,41	7155,28
	100	254	14,41	30,59	15,46	11011,61
40	10	46	10,69	20,41	20,41	8457,81
	20	89	8,59	8,72	8,03	4115,75
	40	187	20,41	15,46	10,59	7673,51
	60	331	8,59	14,41	10,41	5738,95
	100	590	6,73	10,41	14,41	4871,14
60	10	93	10,69	14,41	10,59	6051,00
	20	186	5,4	16,47	6,73	5575,34
	40	347	5,5	15,46	10,69	5575,34
	60	563	10,59	8,72	8,72	4460,35
	100	1143	8,72	14,41	10,59	5772,00
100	10	151	7,08	10,41	20,41	5487,87
	20	405	15,46	8,72	10,41	5302,63
	40	754	4,51	8,72	10,59	3778,69
	60	979	10,59	10,41	10,41	5036,79
	100	1661	5,49	8,72	6,73	3553,78

Fuente: Elaboración propia

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 1 (Análisis final y peso óptimo)

POBLACION	GENERACIONES	TIEMPO [Seg]	AREA FINAL [cm ²]			PESO FINAL	PESO OPTIMO
			A ₁	A ₂	A ₃		
10	10	15	6,54	8,72	8,03	3824,15	3402,58
	20	30	7,08	8,72	7,11	3813,68	
	40	60	5,5	8,72	7,11	3590,48	
	60	79	4,42	8,72	7,11	3437,48	
	100	106	4,42	8,72	6,73	3402,58	
20	10	24	8,03	10,41	10,69	4703,27	3437,48
	20	46	4,42	8,72	7,11	3437,48	
	40	93	4,42	8,72	7,11	3437,48	
	60	137	4,42	8,72	7,11	3437,48	
	100	254	4,42	8,72	7,11	3437,48	
40	10	46	5,5	8,72	7,11	3590,48	3402,58
	20	89	4,42	8,72	7,11	3437,48	
	40	187	4,42	8,72	7,11	3437,48	
	60	331	4,42	8,72	7,11	3437,48	
	100	590	4,42	8,72	6,73	3402,58	
60	10	93	4,42	8,72	10,41	3749,23	3402,58
	20	186	4,42	8,72	6,73	3402,58	
	40	347	4,42	8,72	7,11	3437,48	
	60	563	4,42	8,72	7,11	3437,48	
	100	1143	4,42	8,72	7,11	3437,48	
100	10	151	5,49	8,72	8,03	3676,55	3402,58
	20	405	4,42	8,72	7,11	3437,48	
	40	754	4,42	8,72	6,73	3402,58	
	60	979	4,42	8,72	7,11	3437,48	
	100	1661	4,42	8,72	6,73	3402,58	

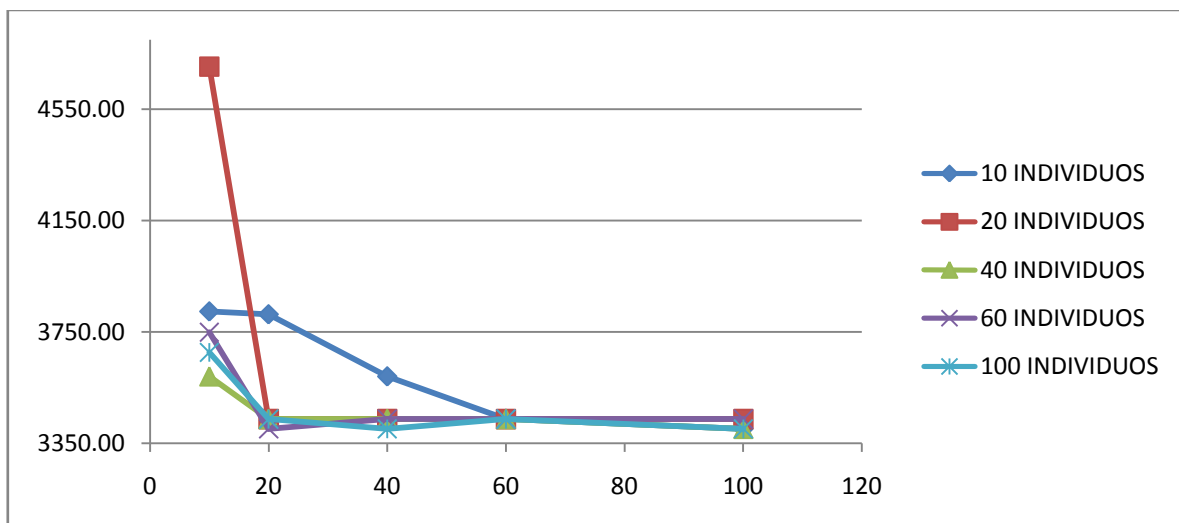
Fuente: Elaboración propia

En la figura 74 se muestra el peso óptimo obtenido para cada población a medida que aumentaban las generaciones. En esta grafica se nota que a medida que se aumenten las generaciones, sin importar el tamaño de población; el peso cada vez más se acerca al diseño óptimo, esto se debe a que entre más individuos haya en la población y mas generaciones se formen el espacio de búsqueda se vuelve más grande y habrá más posibilidades de obtener mejores resultados.

En la figura 75 se muestra el mejor peso óptimo obtenido para cada población. Ya que el algoritmo no se comporta de manera constante en cuanto a la búsqueda del óptimo, podemos notar que en una población no se encontró el peso óptimo pero no quiere decir que el algoritmo falle ya que el valor obtenido para esa población está muy cercano al óptimo.

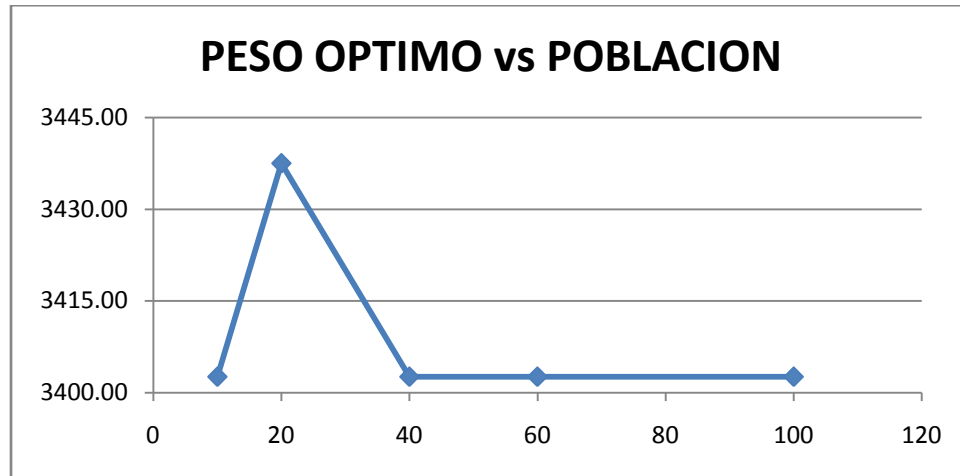
En la figura 76 se muestra el peso óptimo obtenido en cada una de las ejecuciones del algoritmo.

Figura . Peso óptimo obtenido para cada población a medida que aumentaban las generaciones Caso 1.



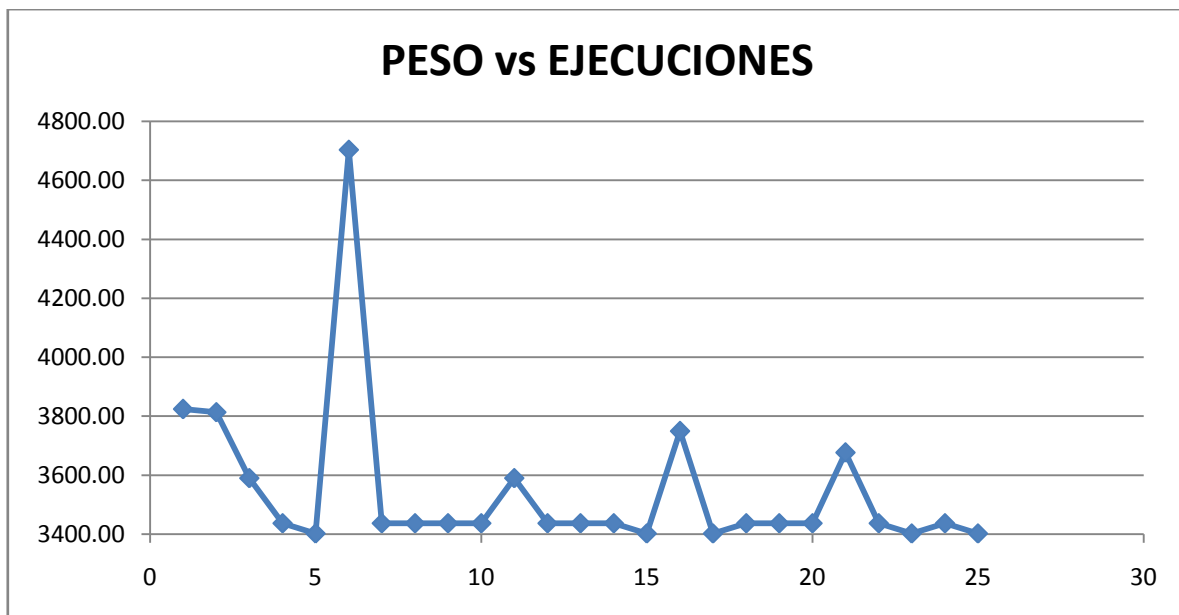
Fuente: Elaboración propia

Figura . Peso óptimo obtenido para cada población Caso1.



Fuente: Elaboración propia

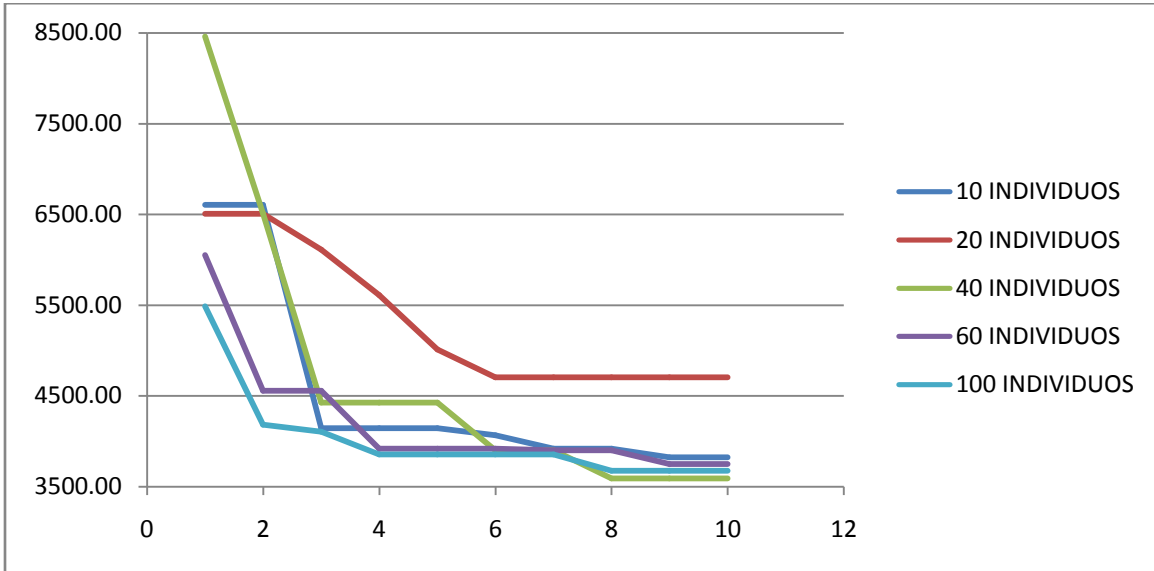
Figura . Peso óptimo obtenido en cada una de las ejecuciones del algoritmo Caso 1



Fuente: Elaboración propia

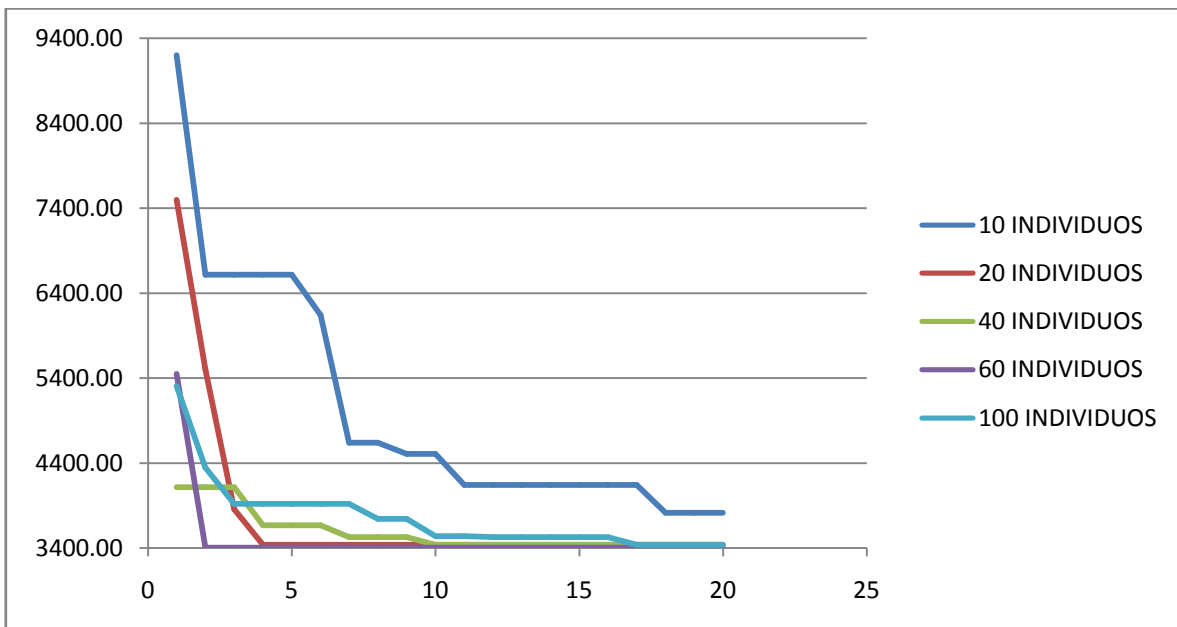
En las figuras 77, 78, 79, 80, 81 se muestra como es la variación del peso cuando se ejecuta el algoritmo para diferentes tamaños de población pero con igual número de generaciones.

Figura . Comparación de la variación del peso con 10 generaciones



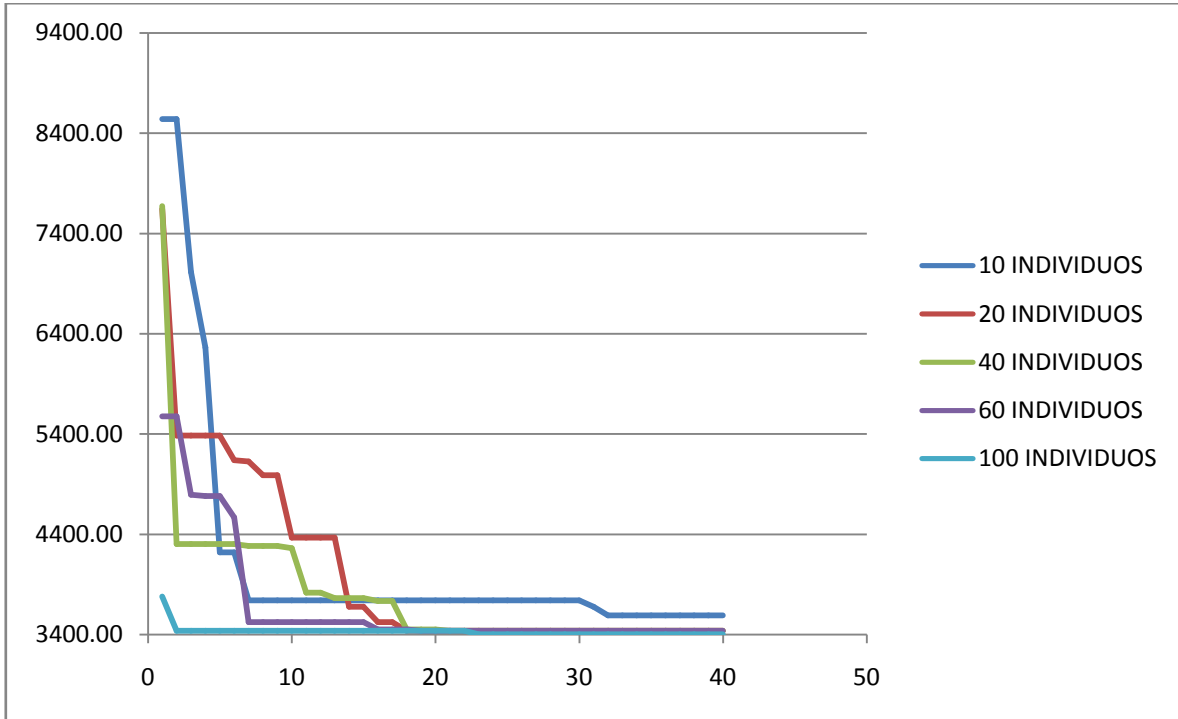
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 20 generaciones



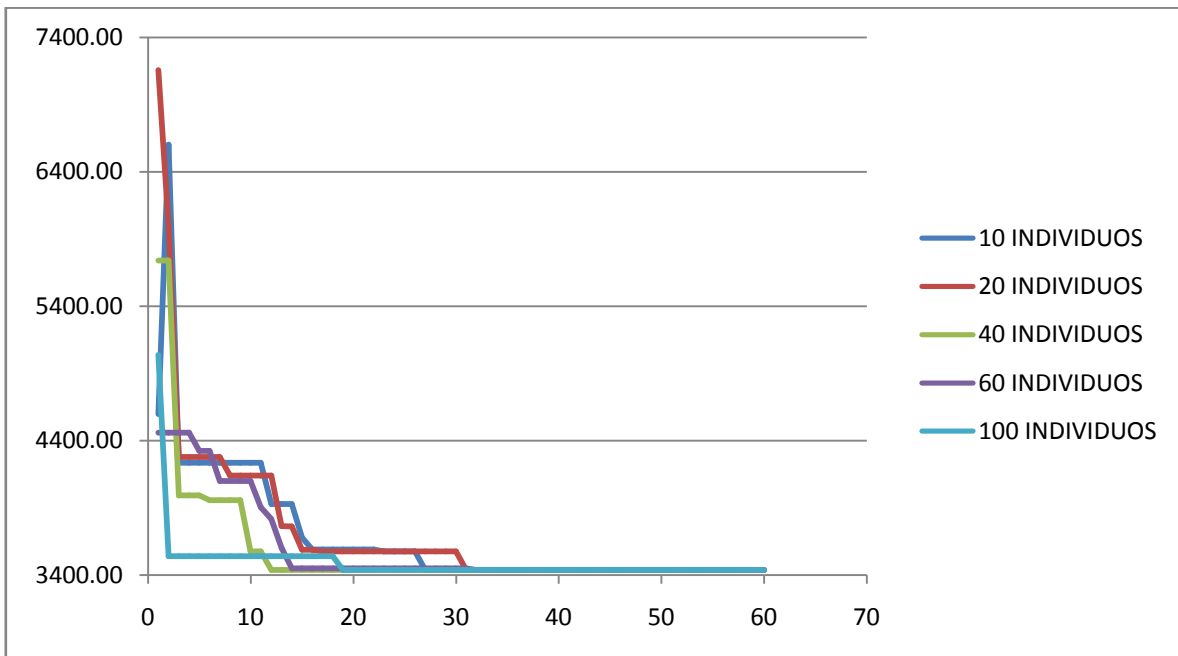
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 40 generaciones



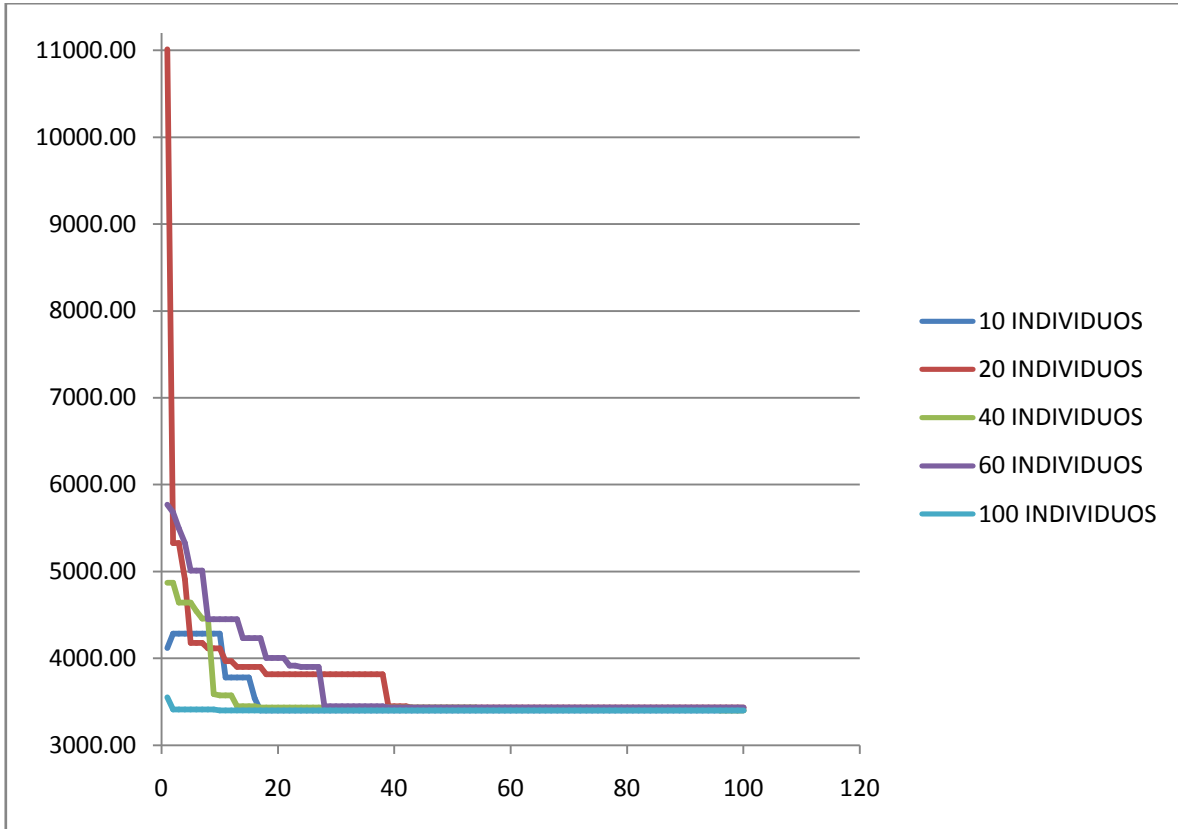
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 60 generaciones



Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 100 generaciones



Fuente: Elaboración propia

De las figuras 77, 78, 79, 80, 81 podemos notar que para todos los casos el primer el individuo generado aleatoriamente por el algoritmo no es una solución buena. Vemos que para 10 generaciones con diferentes tamaños de población no obtenemos buenos resultados.

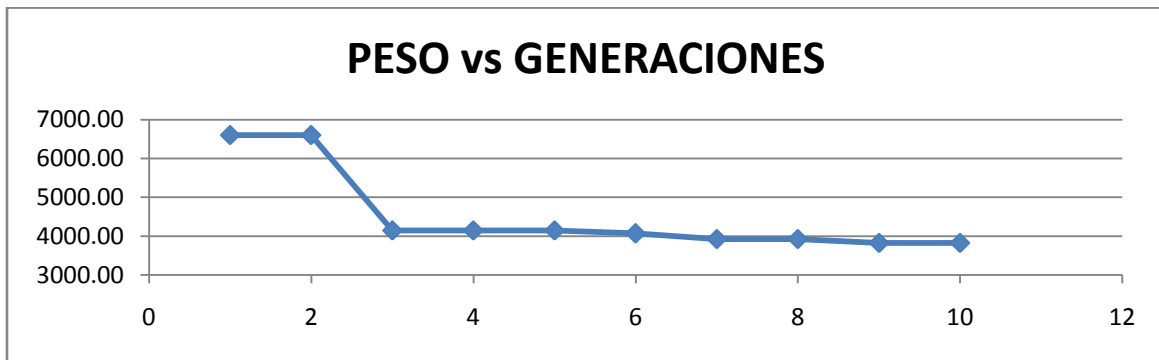
Para 20 generaciones podemos ver que se empiezan a dar mejores resultados aunque para la población de 10 individuos se necesitan más generaciones para poder converger a un óptimo valor del peso o a un valor muy cercano como se puede notar en las demás gráficas para 40, 60, y 100 generaciones.

Lo más notable es que el algoritmo refleja que a medida que se aumenten las generaciones sin importar el tamaño de la población la tendencia es converger a

un punto en común ya sea el mejor individuo con el que se obtiene el peso óptimo de la estructura, o a un punto muy cercano al óptimo. Con lo dicho antes podemos decir que queda demostrado que el algoritmo tiene un buen funcionamiento.

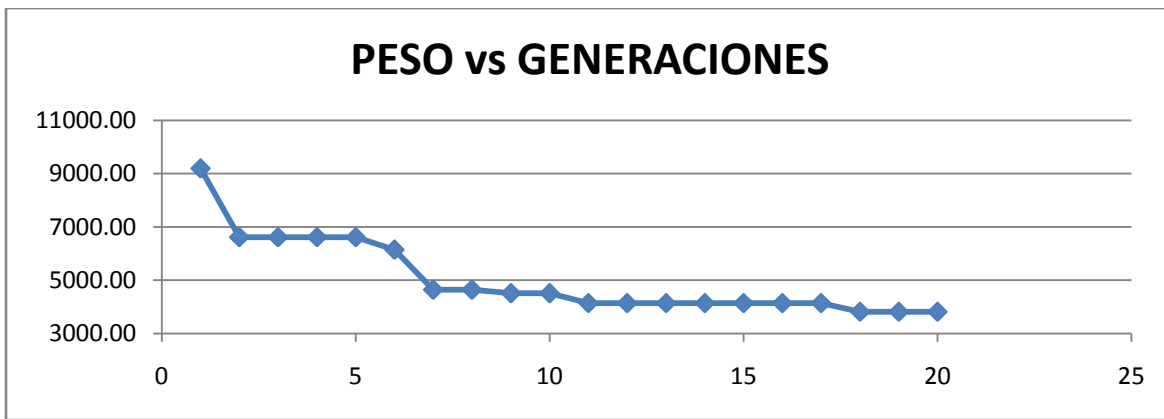
En las siguientes figuras se representa gráficamente la variación del peso óptimo de la estructura en cada generación. Esto se hace para todos los tamaños de población propuestos y para cada una de las ejecuciones del algoritmo con sus respectivas generaciones.

Figura . Variación del peso en cada generación para una población de 10 individuos con 10 generaciones



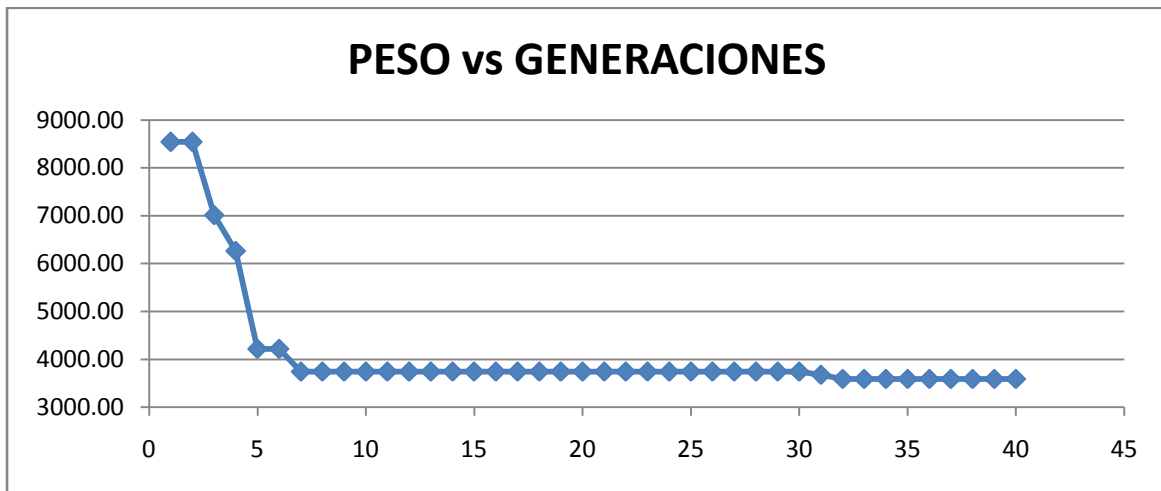
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 10 individuos con 20 generaciones



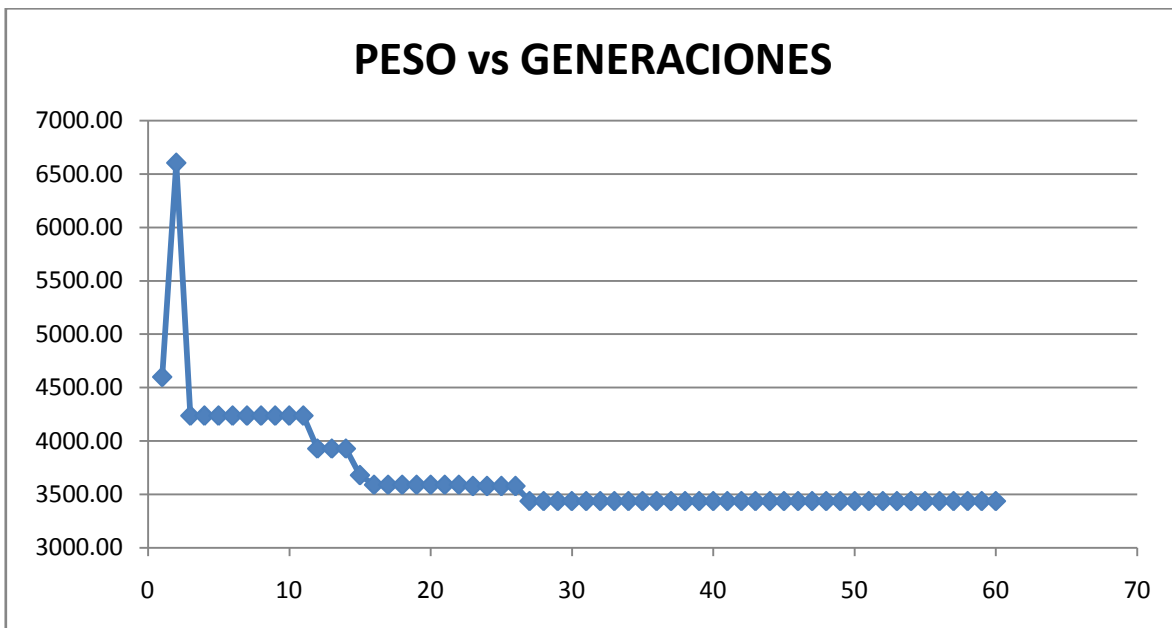
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 10 individuos con 40 generaciones



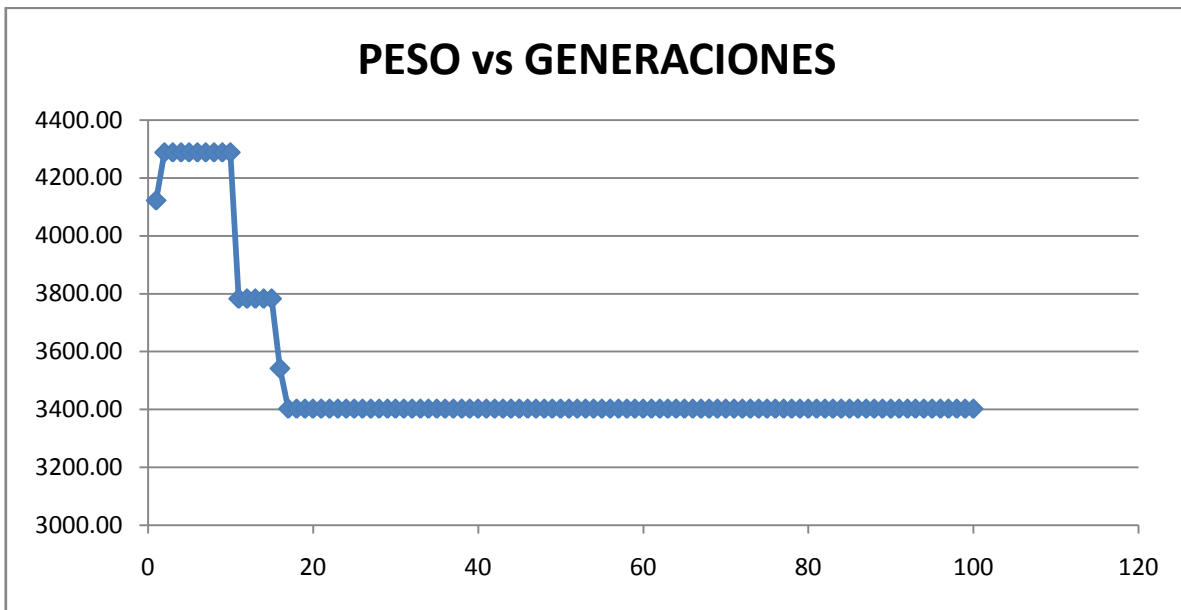
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 10 individuos con 60 generaciones



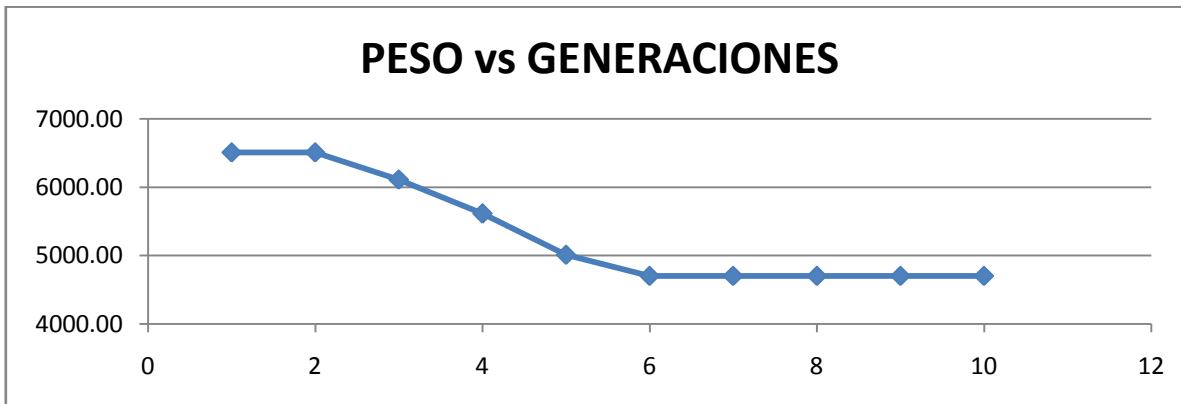
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 10 individuos con 100 generaciones



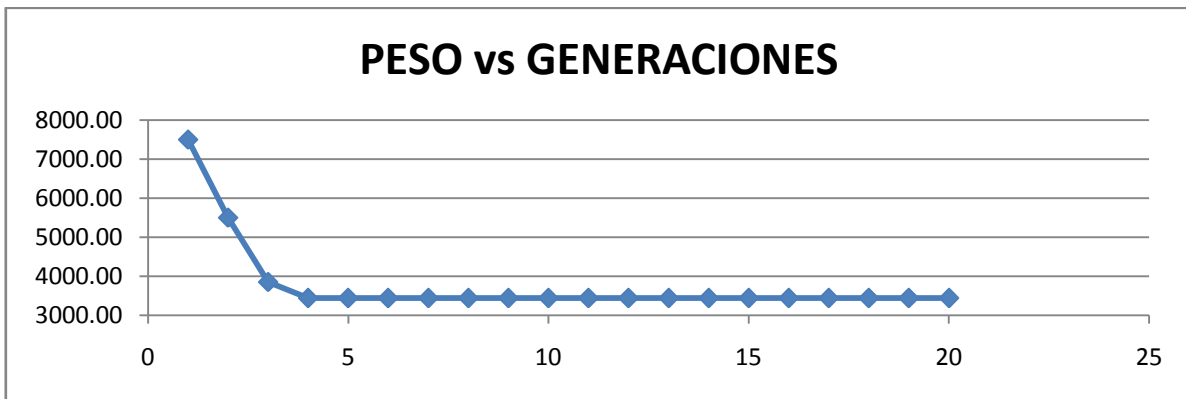
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 10 generaciones



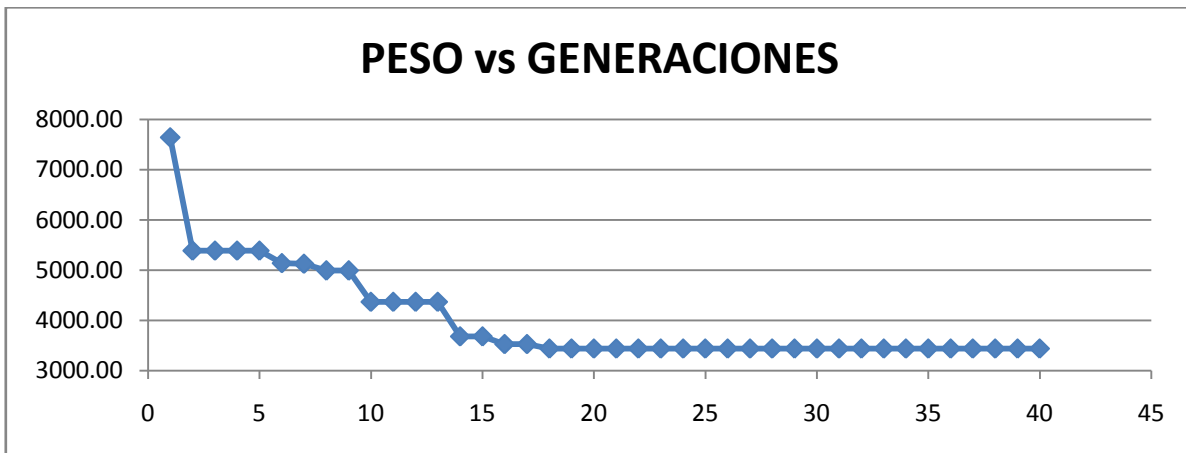
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 20 generaciones



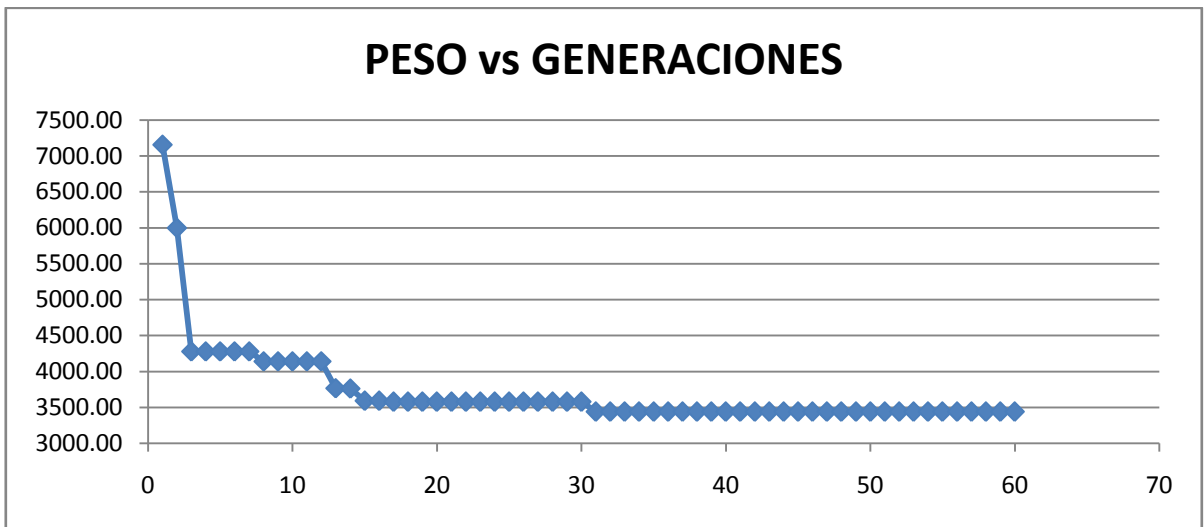
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 40 generaciones



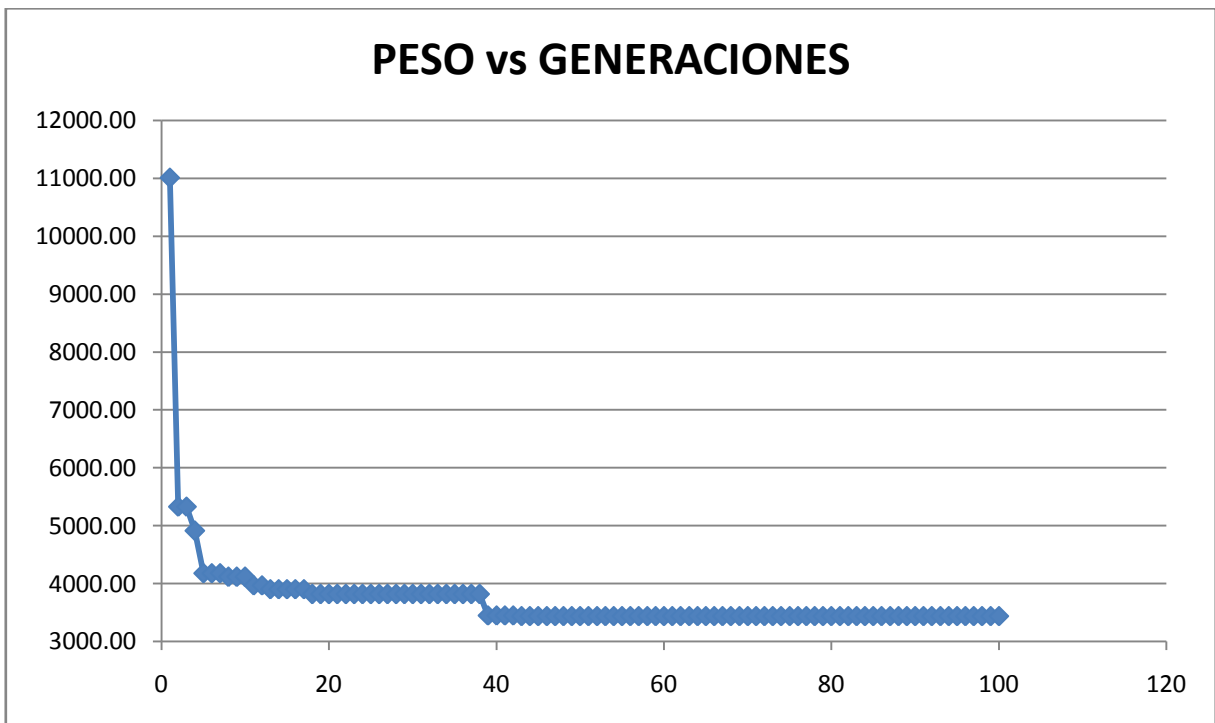
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 60 generaciones



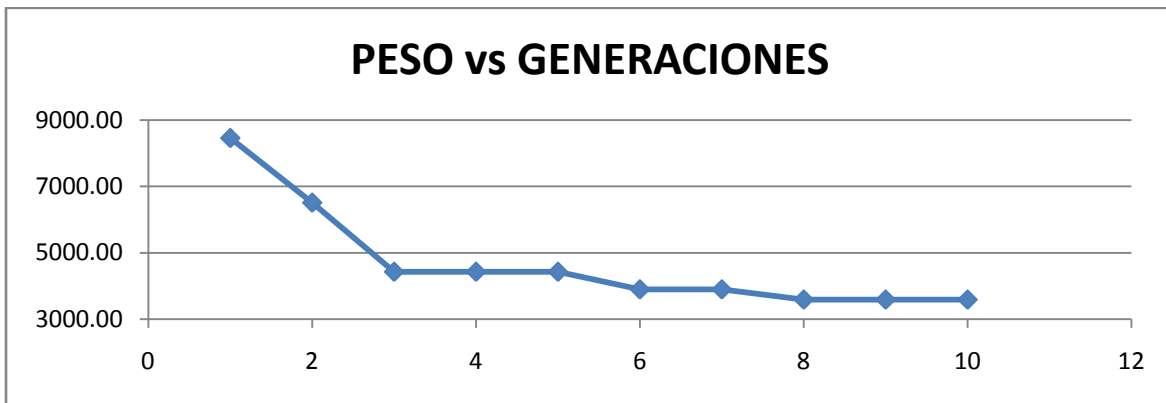
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 100 generaciones



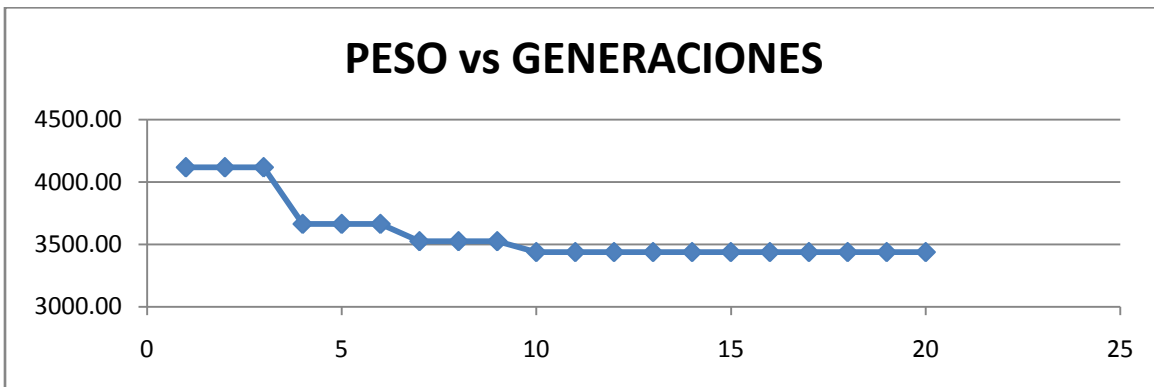
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 40 individuos con 10 generaciones



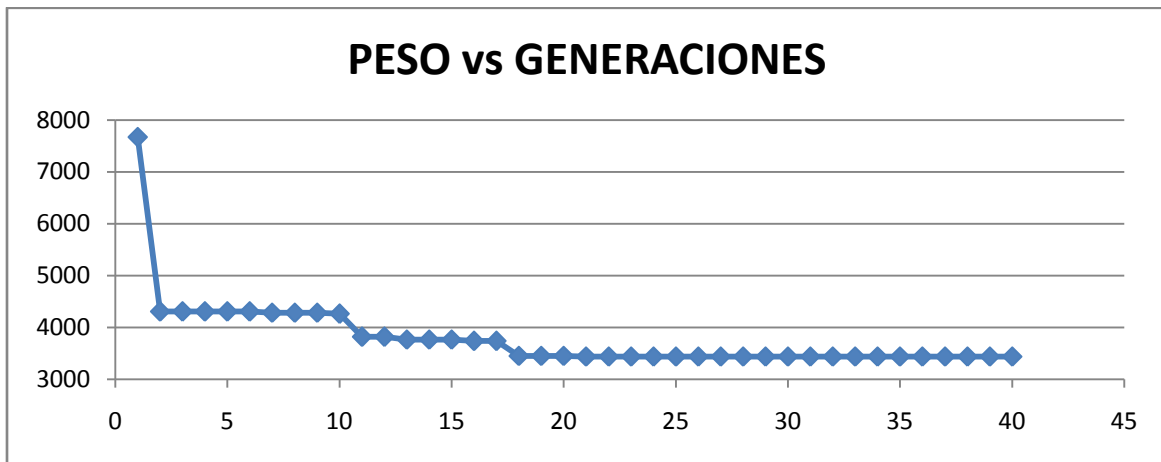
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 40 individuos con 20 generaciones



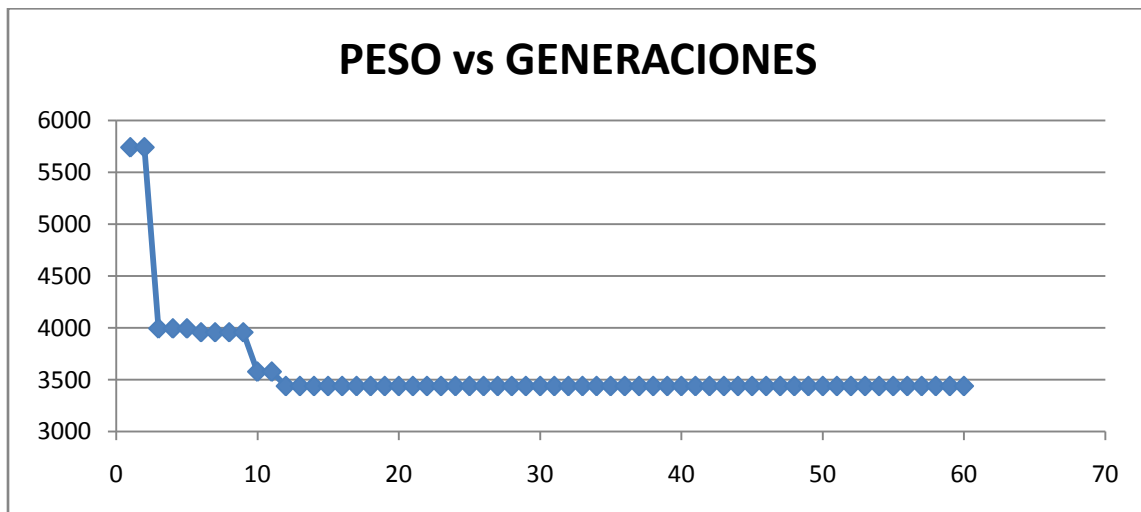
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 40 individuos con 40 generaciones



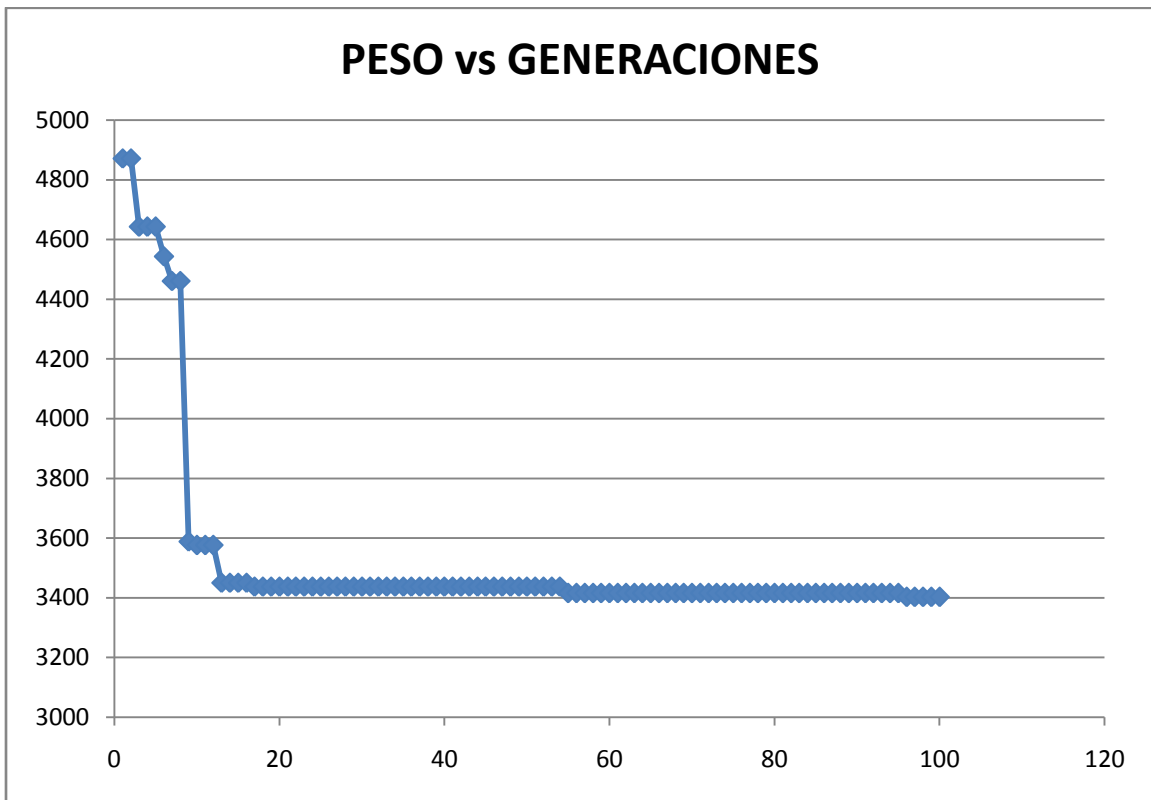
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 40 individuos con 60 generaciones



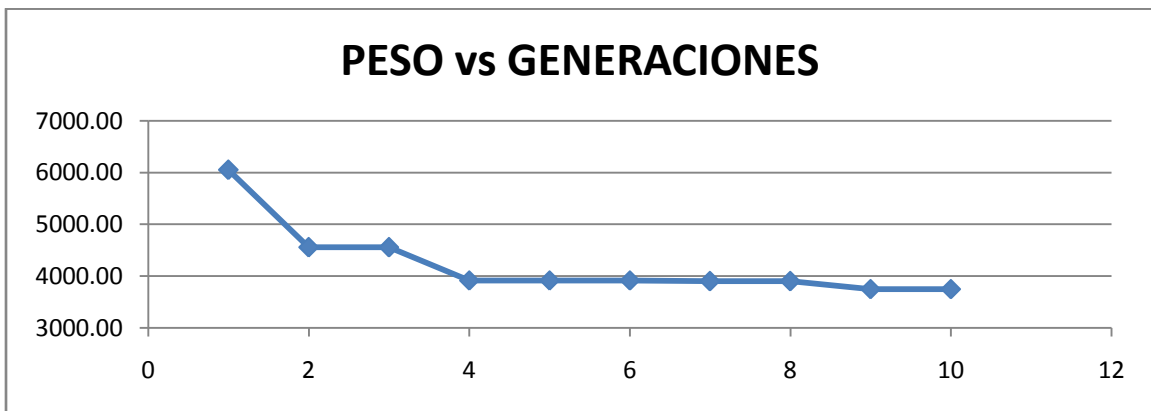
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 40 individuos con 100 generaciones



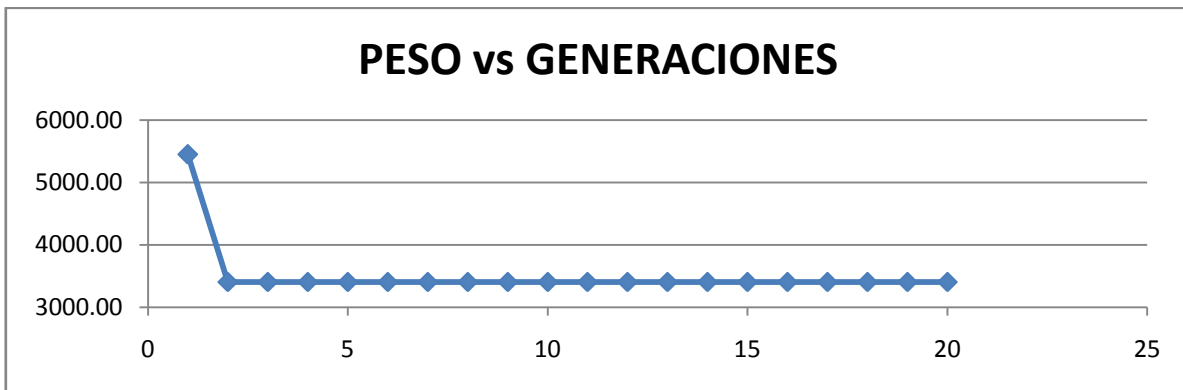
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 60 individuos con 10 generaciones



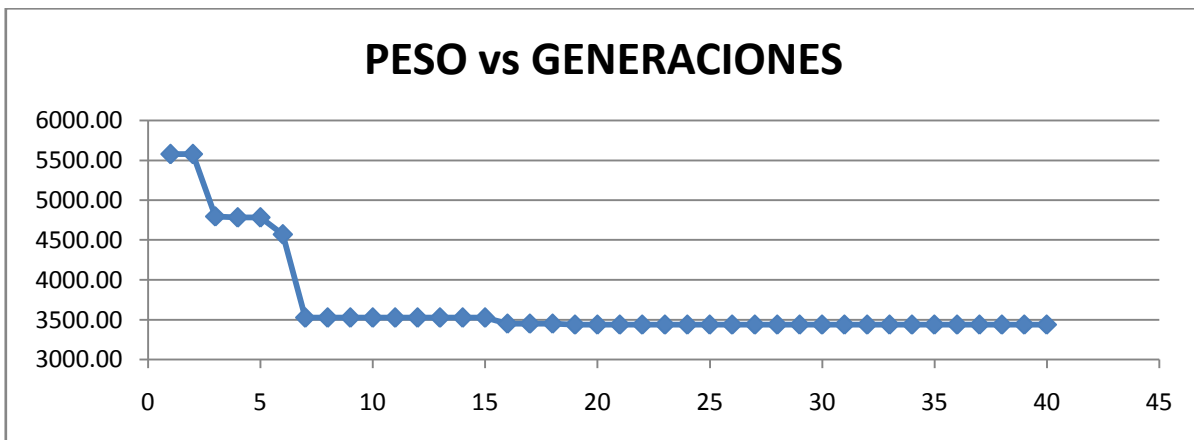
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 60 individuos con 20 generaciones



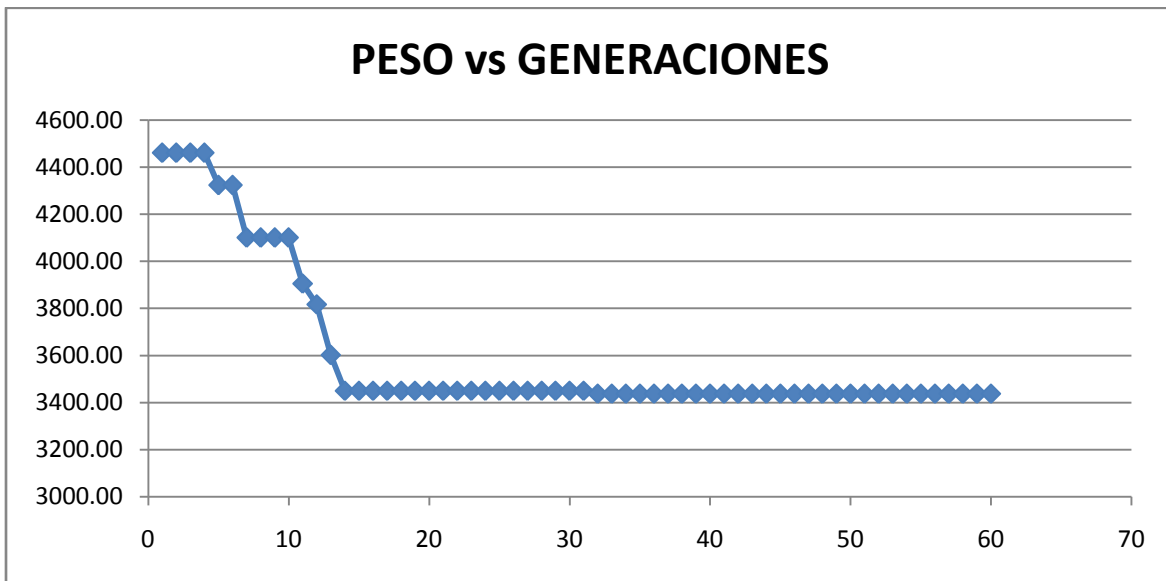
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 60 individuos con 40 generaciones



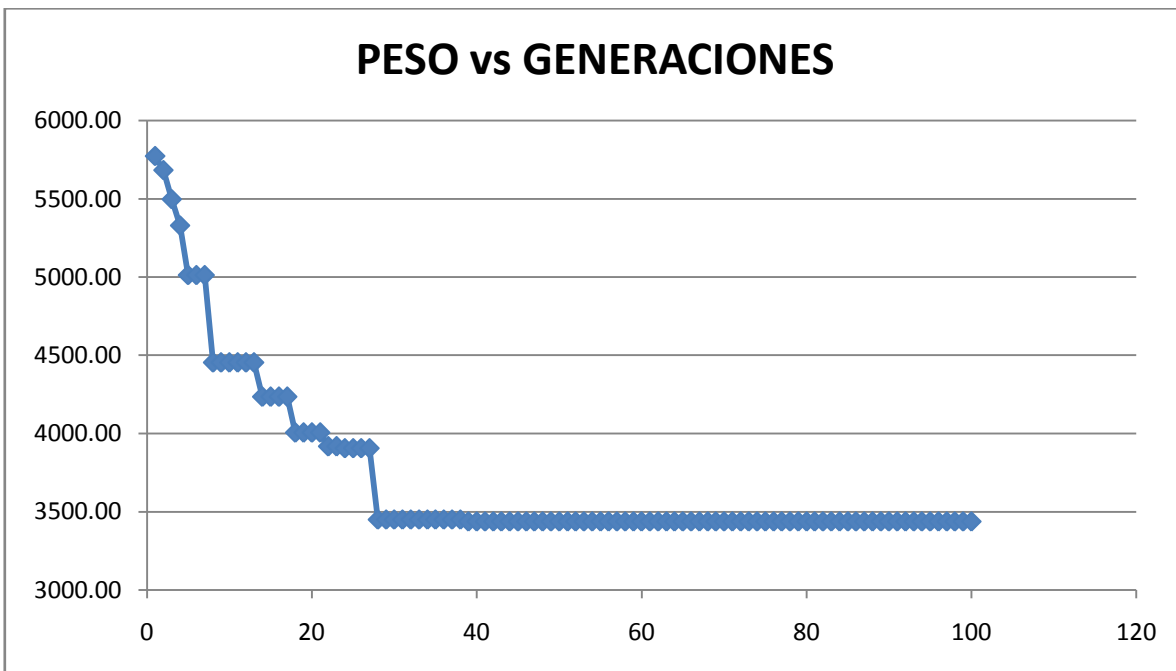
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 60 individuos con 60 generaciones



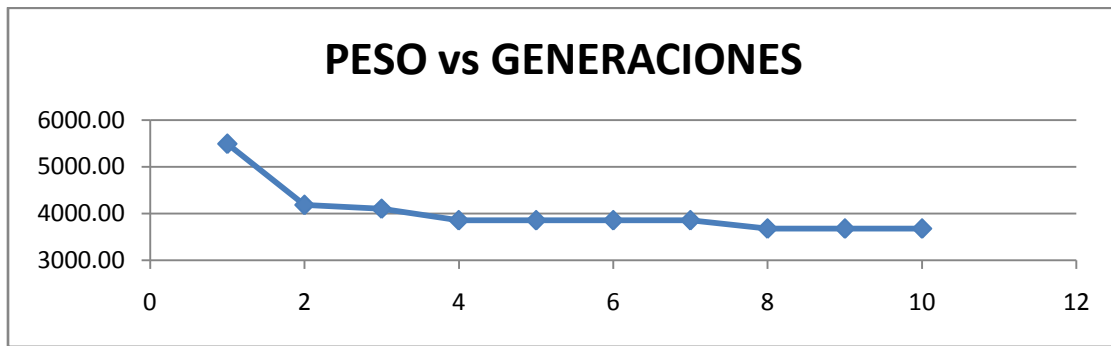
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 60 individuos con 100 generaciones



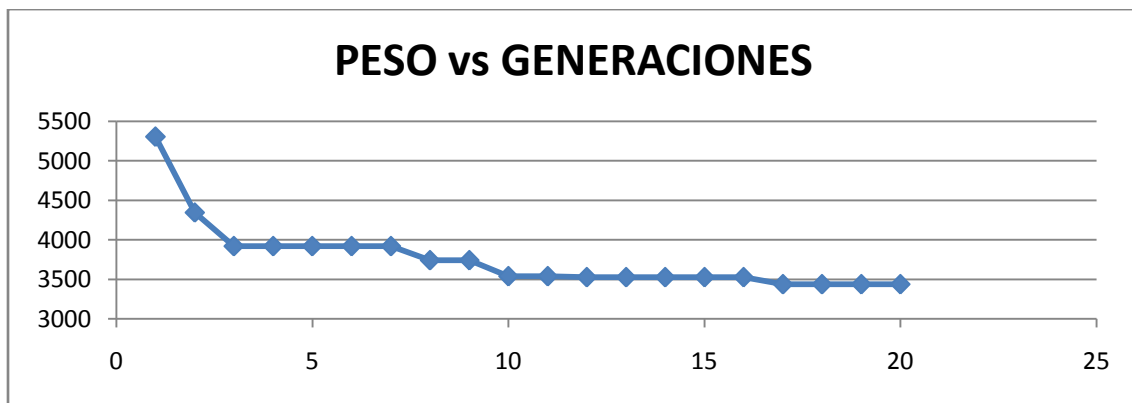
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 100 individuos con 10 generaciones



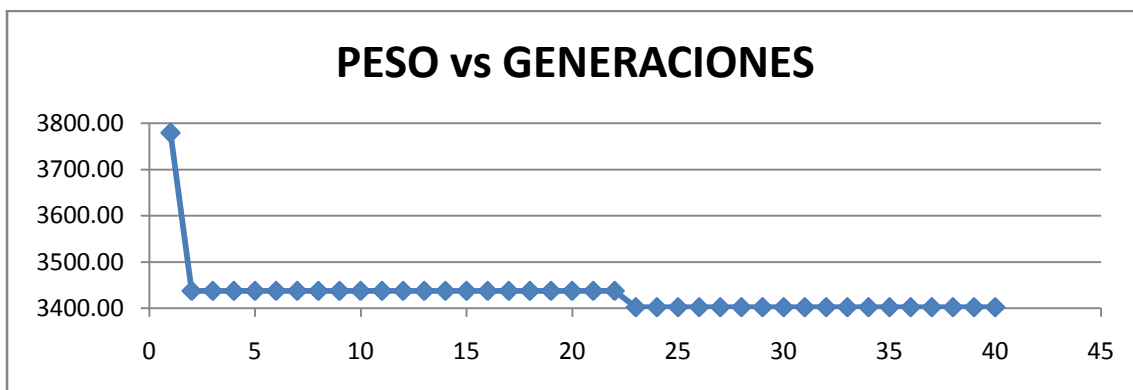
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 100 individuos con 20 generaciones



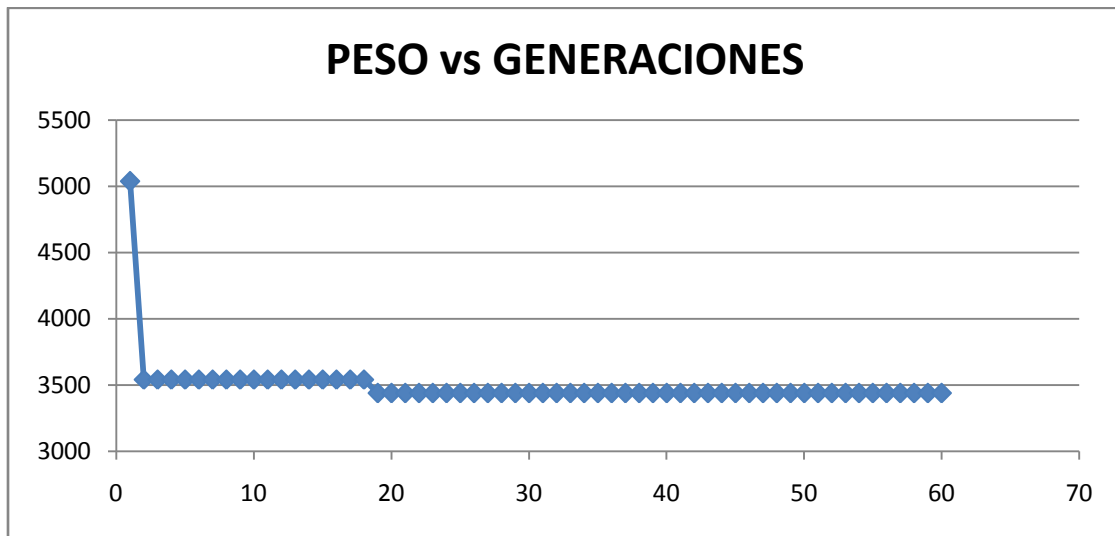
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 100 individuos con 40 generaciones



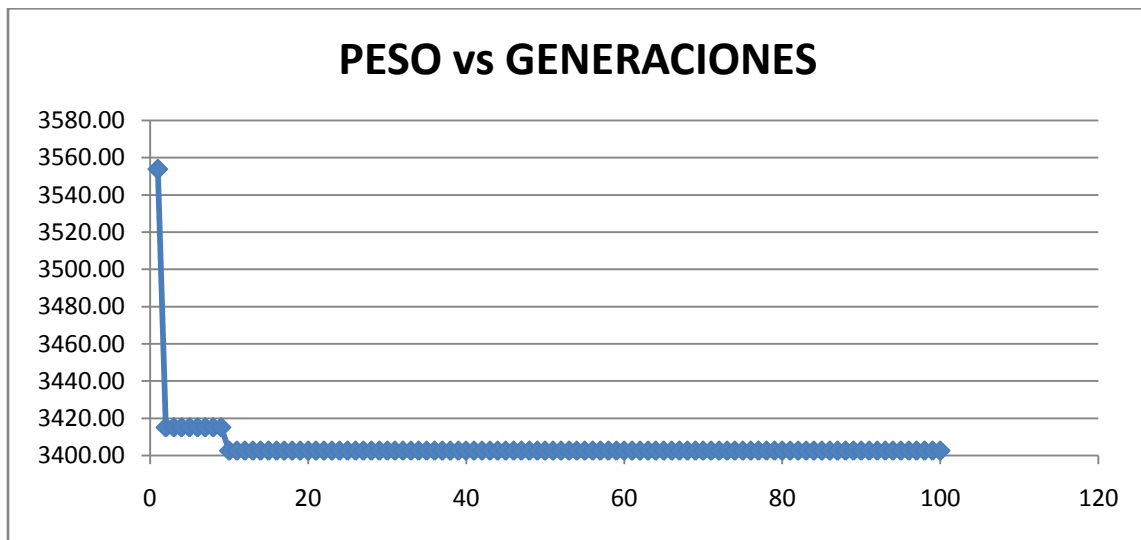
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 100 individuos con 60 generaciones



Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 100 individuos con 100 generaciones



Fuente: Elaboración propia

Una vez obtenidos los resultados podemos notar que la optimización se lleva a cabo satisfactoriamente, aunque no siempre se obtuvo el resultado esperado, es decir el resultado ya obtenido antes con el algoritmo PSO. Al menos en un número de generaciones propuesto para cada población se obtuvo el diseño óptimo, con excepción un tamaño de población que se obtuvo un valor muy cercano.

A medida que se aumenta el número de generaciones y el tamaño de la población se aumenta el tiempo de ejecución del algoritmo, aumentando también el espacio de búsqueda y así el algoritmo puede explorar más a fondo y encontrar valores óptimos.

En la tabla 21 se hace una comparación del peso obtenido en un diseño inicial propuesto por los autores, además podemos ver que el diseño óptimo obtenido por el algoritmo genético AG, el resultado es similar al obtenido por el PSO.

Tabla Comparación del diseño inicial y el diseño óptimo obtenido con el AG.

DISEÑO	A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	PESO	%
INICIAL	4,51	10,41	7,11	3866,43	REDUCCION DEL PESO
OPTIMO	4,42	8,72	6,73	3402,58	12,00

Fuente: Elaboración propia

4.3.2 Resultados para el caso 2

Para la agrupación de miembros del caso 2, se hicieron pruebas para cinco poblaciones de 20, 50, 80, 120 y 150 individuos cada una, para las tres primeras poblaciones se hicieron pruebas con 20, 40, 60, 100 y 150 generaciones, y para las poblaciones de 120 y 150 individuos se hicieron pruebas con 20, 40, 60, y 100 generaciones cada una.

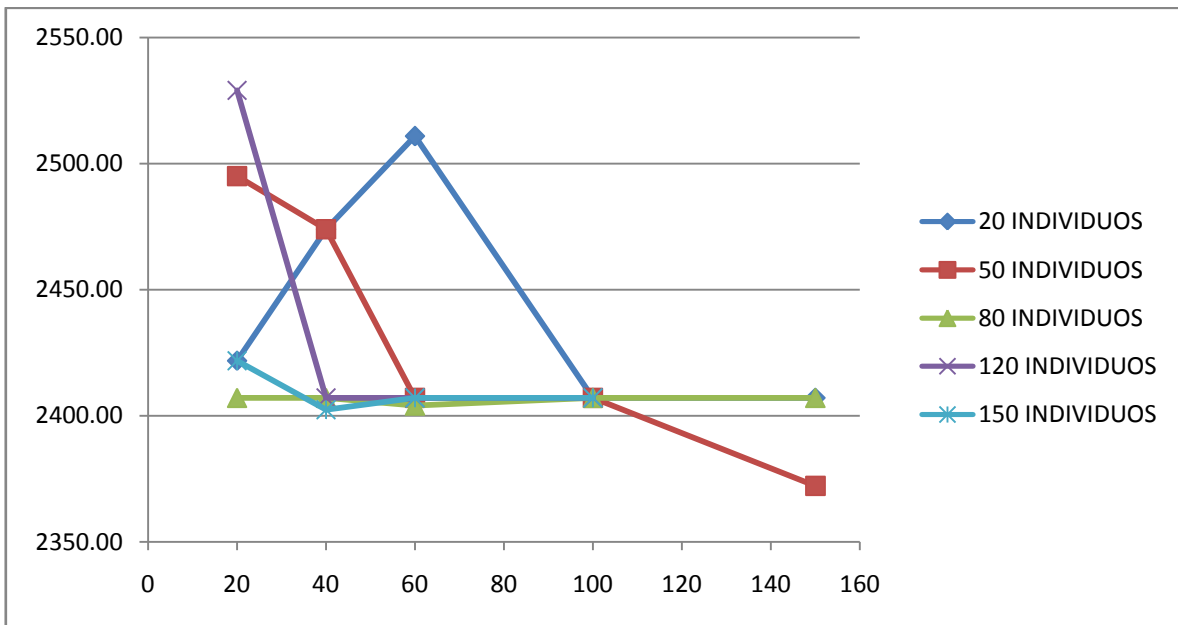
En la tabla 22 se muestra un resumen de los resultados obtenidos después de realizar las ejecuciones del algoritmo, se puede observar el tamaño de la población, las generaciones en cada prueba, las cuatro áreas iniciales escogidas aleatoriamente por el algoritmo, el peso de la cercha al iniciar el algoritmo, luego tenemos las cuatro áreas finales es decir una vez terminada la optimización, luego encontramos el peso óptimo de la estructura, y por último tenemos el mejor peso para cada población.

En la figura 107 se muestra el peso óptimo obtenido para cada población a medida que aumentaban las generaciones. En esta gráfica se nota que a medida que se aumenten las generaciones, sin importar el tamaño de población; el peso cada vez más se acerca al diseño óptimo, esto se debe a que entre más individuos haya en la población y más generaciones se formen el espacio de búsqueda se vuelve más grande y habrá más posibilidades de obtener mejores resultados. Aun así solo una población en una de sus generaciones obtuvo el peso óptimo obtenido con el algoritmo PSO. Todos los demás ensayos se dirigían a un diseño muy cercano al óptimo.

En la figura 108 se muestra el peso óptimo obtenido para cada población. Ya que el algoritmo no se comporta de manera constante en cuanto a la búsqueda del óptimo, podemos notar que en tan solo una población se encontró el peso óptimo pero no quiere decir que el algoritmo falle ya que los valores obtenidos en la demás poblaciones está muy cercano al óptimo esperado.

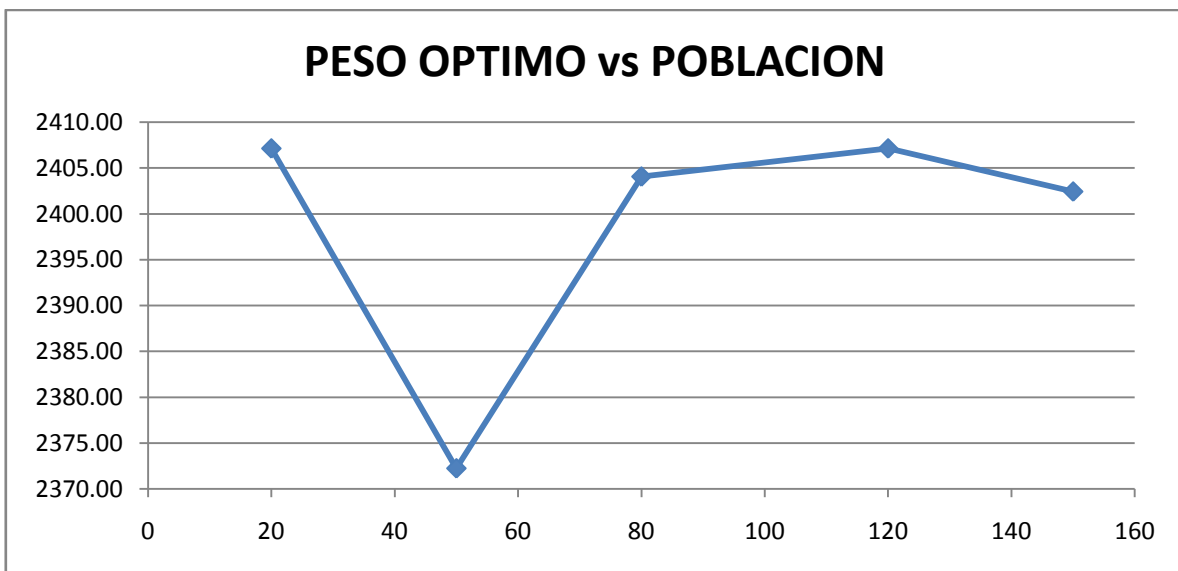
En la figura 109 se muestra el peso óptimo obtenido en cada una de las ejecuciones del algoritmo.

Figura . Peso óptimo obtenido para cada población a medida que aumentaban las generaciones Caso 2.



Fuente: Elaboración propia

Figura . Peso óptimo obtenido para cada población Caso 2.



Fuente: Elaboración propia

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 2 (Análisis inicial)

POBLACION	GENERACIONES	TIEMPO [Seg]	AREA INICIAL [cm ²]				PESO INICIAL
			A ₁	A ₂	A ₃	A ₄	
20	20	75	8,03	36,01	6,73	15,46	10433,78
	40	149	15,46	5,5	7,11	36,01	4481,31
	60	224	10,41	4,42	33,63	25,65	5928,97
	100	378	5,49	6,73	8,03	13,76	3255,59
	150	567	33,63	5,49	30,59	20,41	9129,13
50	20	191	5,5	8,72	13,76	7,11	4204,11
	40	384	4,51	10,69	6,73	14,41	3934,45
	60	562	6,54	6,54	10,69	10,59	3576,29
	100	945	15,46	6,54	6,73	20,41	4547,78
	150	1412	7,11	8,72	6,73	13,76	3828,56
80	20	300	5,4	4,42	7,11	30,59	2760,77
	40	601	10,41	6,54	13,76	20,41	4503,61
	60	887	4,51	4,42	10,41	33,63	2974,28
	100	1509	6,54	7,11	8,72	33,63	3736,31
	150	3146	8,72	5,5	16,47	20,41	4276,47
120	20	454	8,72	7,11	7,11	30,59	3864,69
	40	887	5,4	7,11	6,73	7,11	3146,74
	60	1367	5,4	8,59	10,69	16,47	3958,21
	100	1800	10,59	5,5	16,47	20,41	4541,07
150	20	598	5,5	10,41	6,73	15,46	4017,76
	40	1067	5,4	4,42	6,73	7,11	2510,82
	60	1360	4,51	6,54	15,46	7,11	3704,82
	100	2300	8,59	4,42	10,59	33,63	3568,93

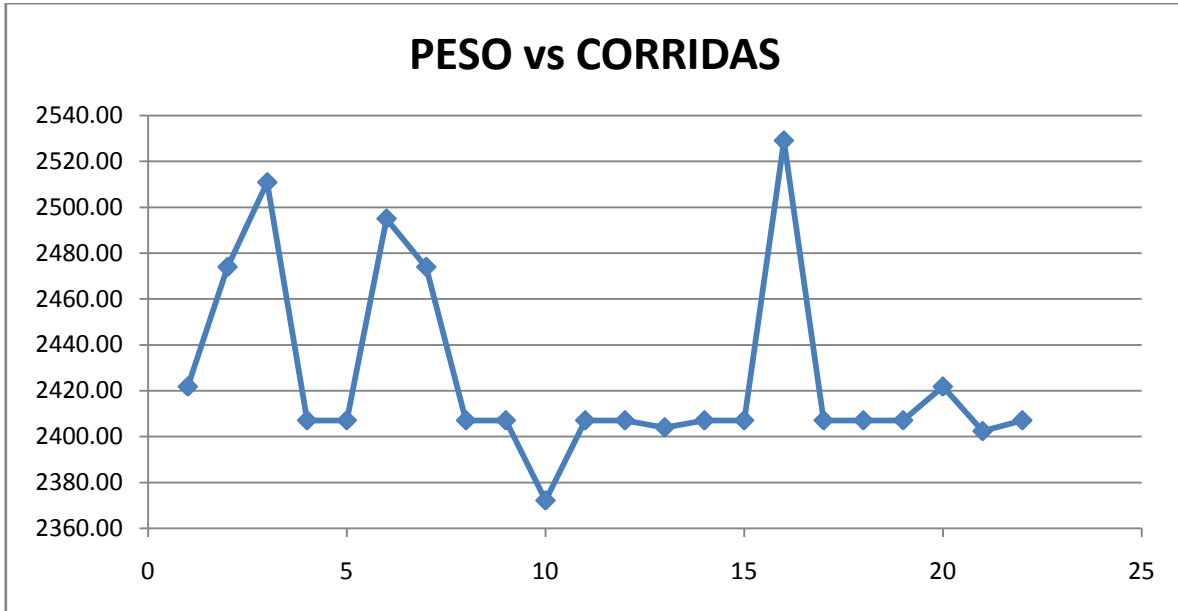
Fuente: Elaboración propia

Tabla Resumen de los resultados obtenidos para la ejecución del algoritmo genético para el Caso 2 (Análisis final y peso óptimo)

POBLACION	GENERACIONES	TIEMPO [Seg]	AREA FINAL [cm ²]				PESO FINAL	PESO OPTIMO
			A ₁	A ₂	A ₃	A ₄		
20	20	75	4,42	4,42	7,11	8,72	2421,82	2407,13
	40	149	4,42	4,42	7,11	14,41	2473,95	
	60	224	5,4	4,42	6,73	7,11	2510,82	
	100	378	4,42	4,42	7,11	7,11	2407,13	
	150	567	4,42	4,42	7,11	7,11	2407,13	
50	20	191	4,42	4,42	8,03	7,11	2494,99	2372,22
	40	384	4,42	4,42	7,11	14,41	2473,95	
	60	562	4,42	4,42	7,11	7,11	2407,13	
	100	945	4,42	4,42	7,11	7,11	2407,13	
	150	1412	4,42	4,42	6,73	7,11	2372,22	
80	20	300	4,42	4,42	7,11	7,11	2407,13	2404,06
	40	601	4,42	4,42	7,11	7,11	2407,13	
	60	887	4,42	4,42	6,73	10,59	2404,06	
	100	1509	4,42	4,42	7,11	7,11	2407,13	
	150	3146	4,42	4,42	7,11	7,11	2407,13	
120	20	454	4,42	4,42	7,11	20,41	2528,99	2407,13
	40	887	4,42	4,42	7,11	7,11	2407,13	
	60	1367	4,42	4,42	7,11	7,11	2407,13	
	100	1800	4,42	4,42	7,11	7,11	2407,13	
150	20	598	4,42	4,42	7,11	8,72	2421,82	2402,42
	40	1067	4,42	4,42	6,73	10,41	2402,42	
	60	1360	4,42	4,42	7,11	7,11	2407,13	
	100	2300	4,42	4,42	7,11	7,11	2407,13	

Fuente: Elaboración propia

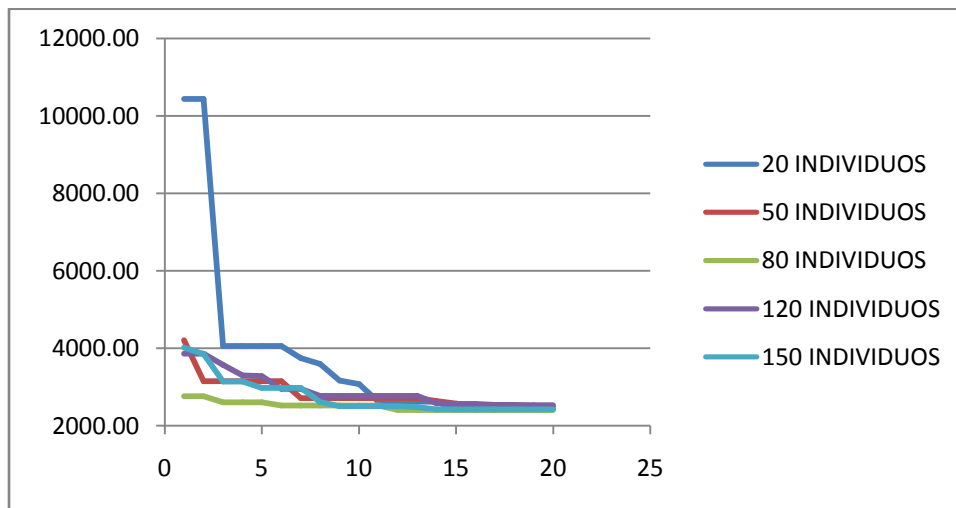
Figura . Peso óptimo obtenido en cada una de las ejecuciones del algoritmo Caso 2.



Fuente: Elaboración propia

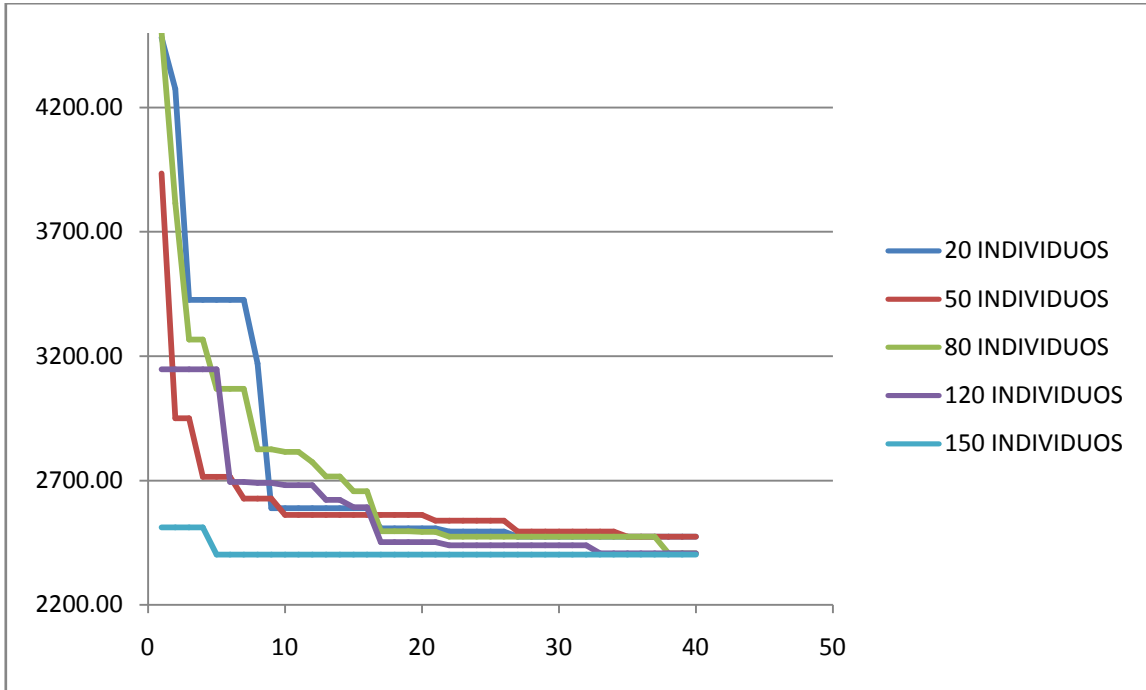
En las figuras 110, 111, 112, 113, 114 se muestra como es la variación del peso cuando se ejecuta el algoritmo para diferentes tamaños de población pero con igual número de generaciones.

Figura . Comparación de la variación del peso con 20 generaciones (Caso 2)



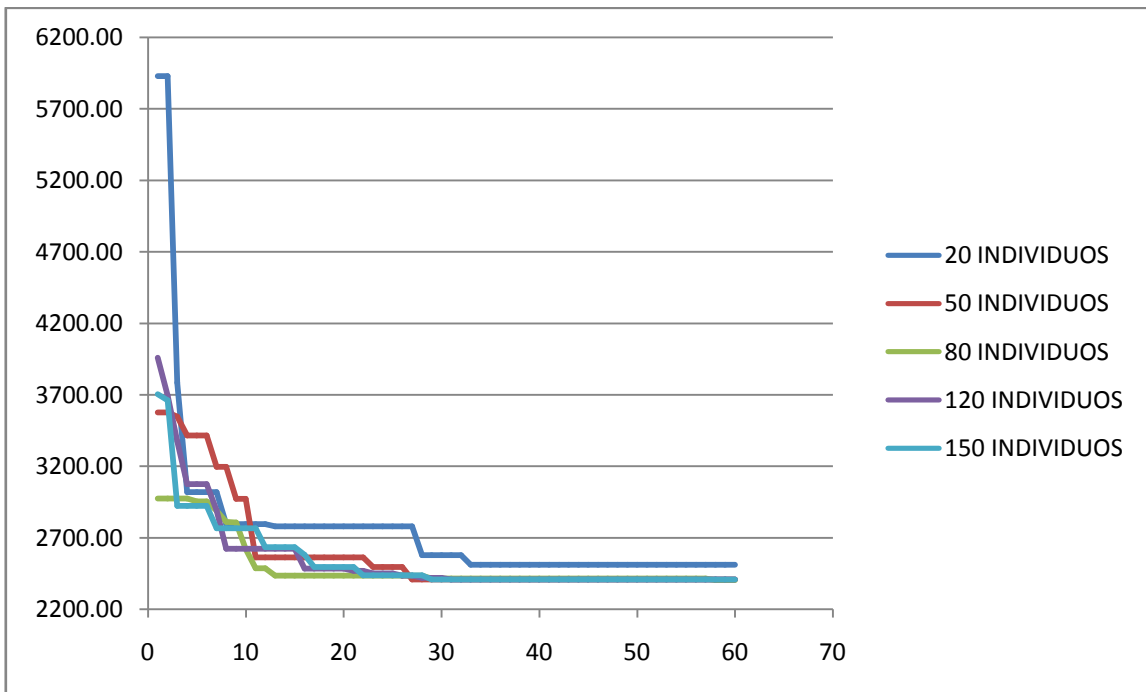
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 40 generaciones (Caso 2)



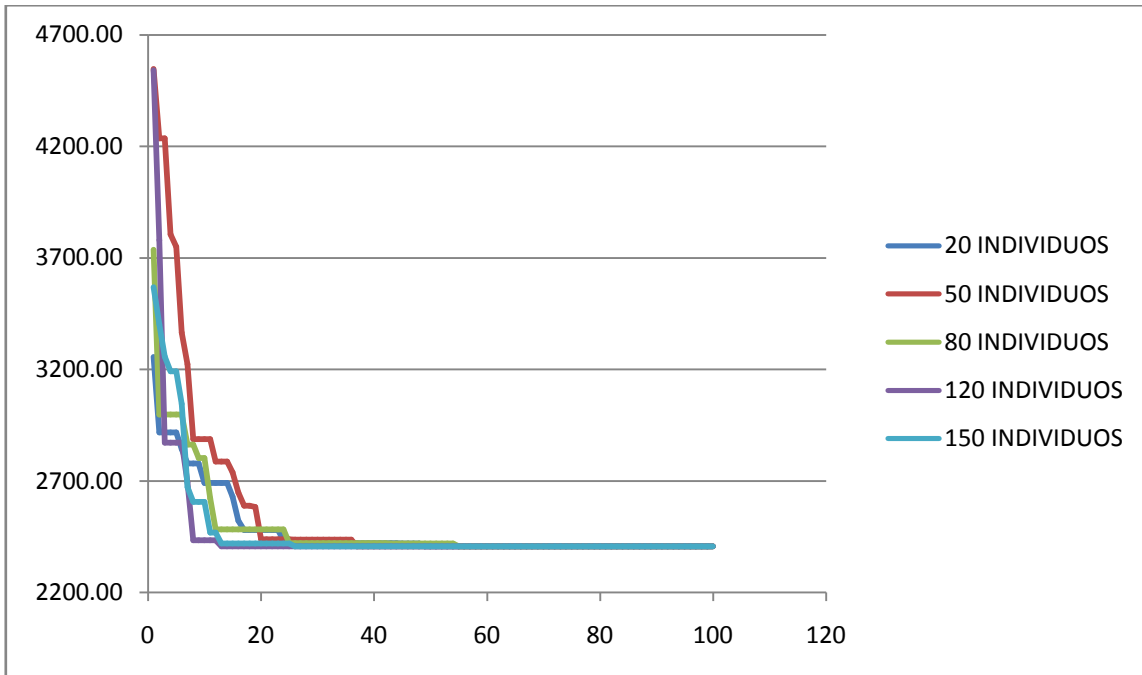
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 60 generaciones



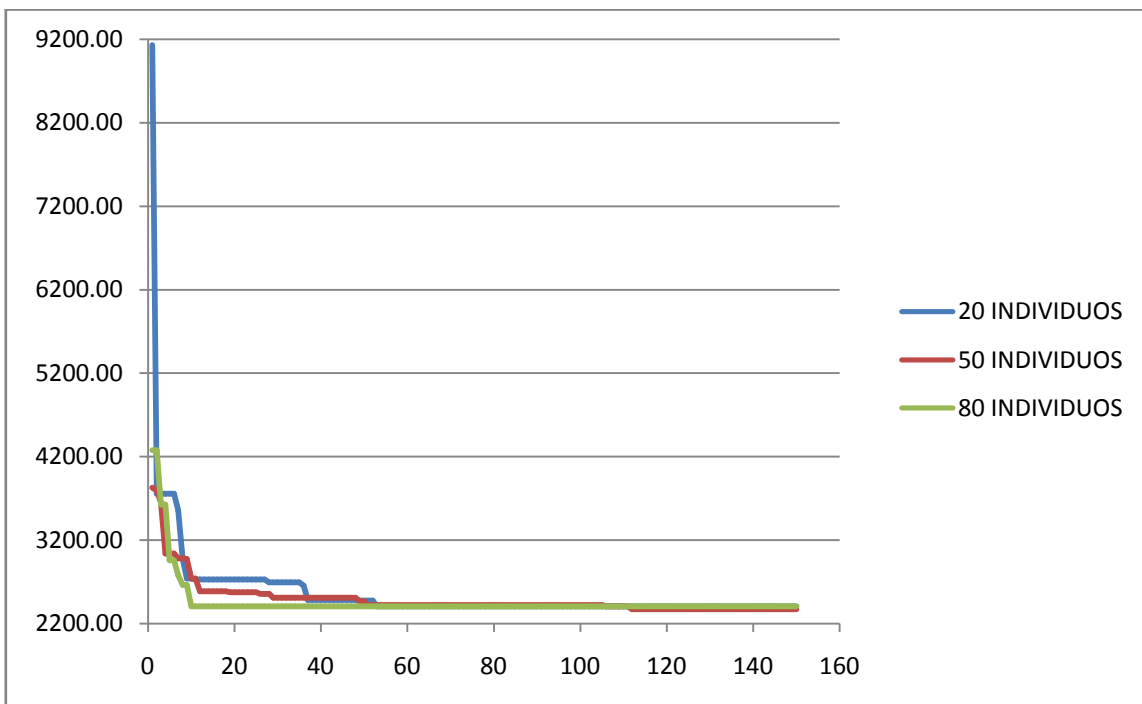
Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 100 generaciones



Fuente: Elaboración propia

Figura . Comparación de la variación del peso con 150 generaciones



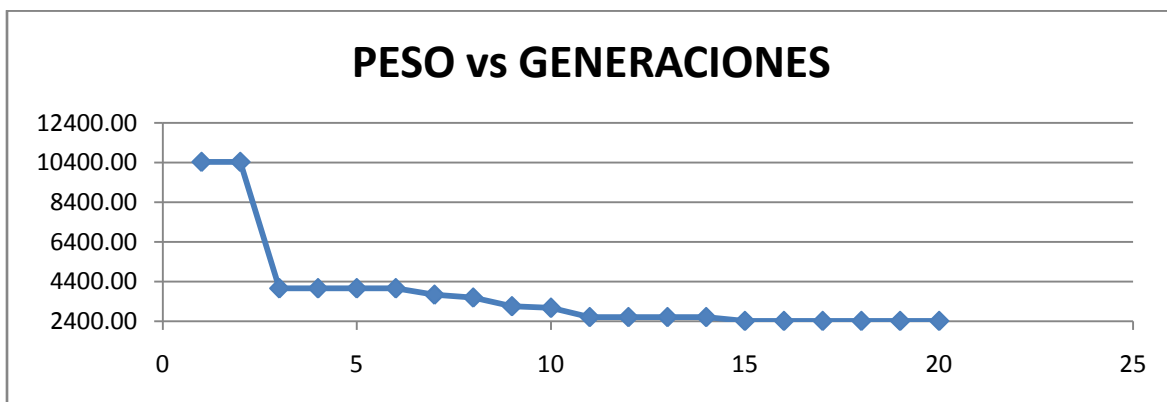
Fuente: Elaboración propia

Al igual que en el Caso 1, notamos que el primer individuo no es una buena solución al problema.

A diferencia que para la optimización del Caso 1, vemos que el comportamiento del mejor individuo para diferentes tamaños de población y con igual número de generaciones es converger a un peso muy cercano al óptimo. Tan solo en la población de 50 individuos y para 150 generaciones el algoritmo llegó al punto donde el peso es el óptimo, igualando la optimización hecha ya por el algoritmo PSO.

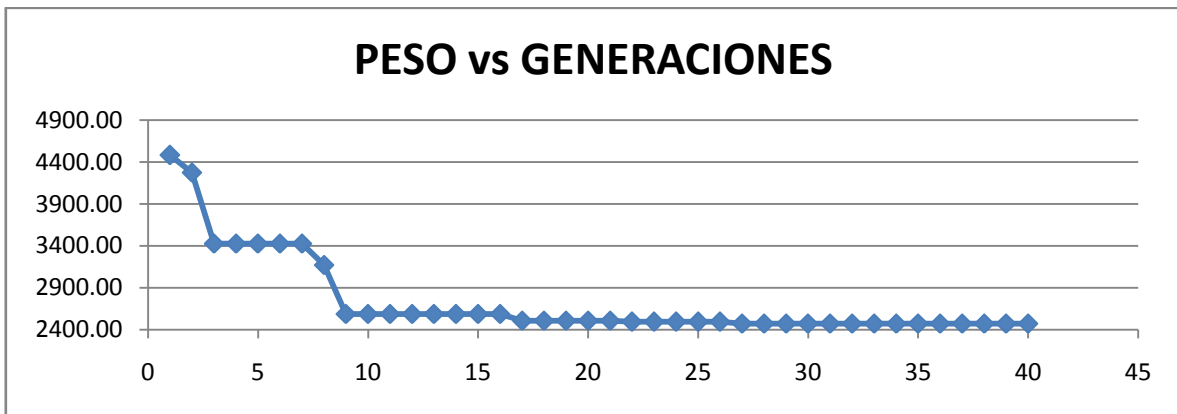
En las siguientes figuras se representa gráficamente la variación del peso óptimo de la estructura en cada generación. Esto se hace para todos los tamaños de población propuestos y para cada una de las ejecuciones del algoritmo con sus respectivas generaciones.

Figura . Variación del peso en cada generación para una población de 20 individuos con 20 generaciones



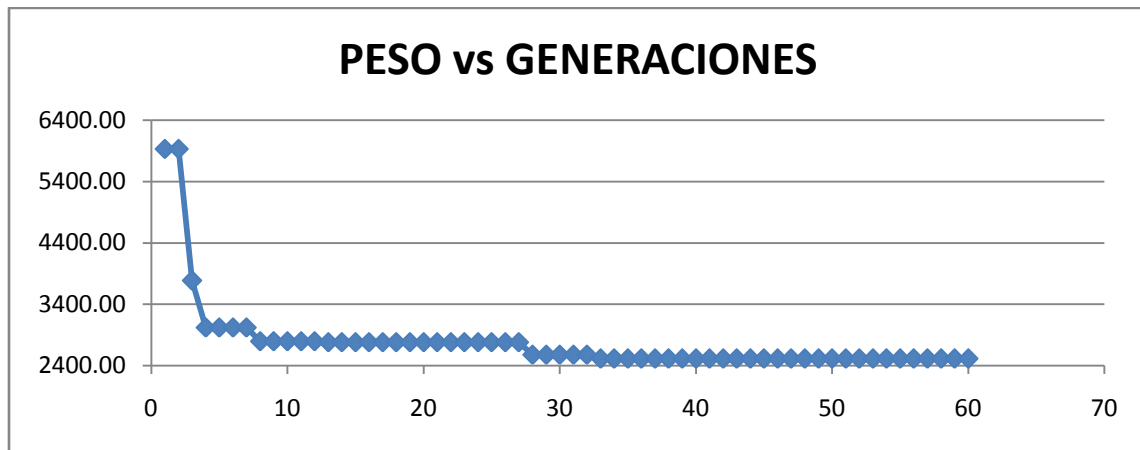
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 40 generaciones



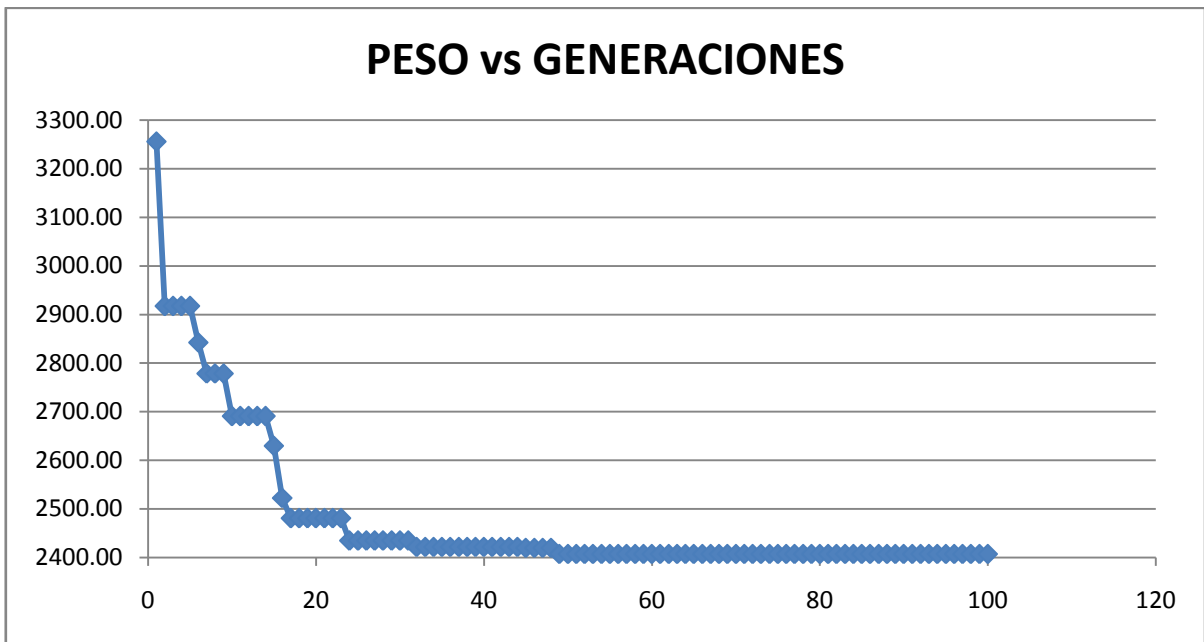
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 60 generaciones



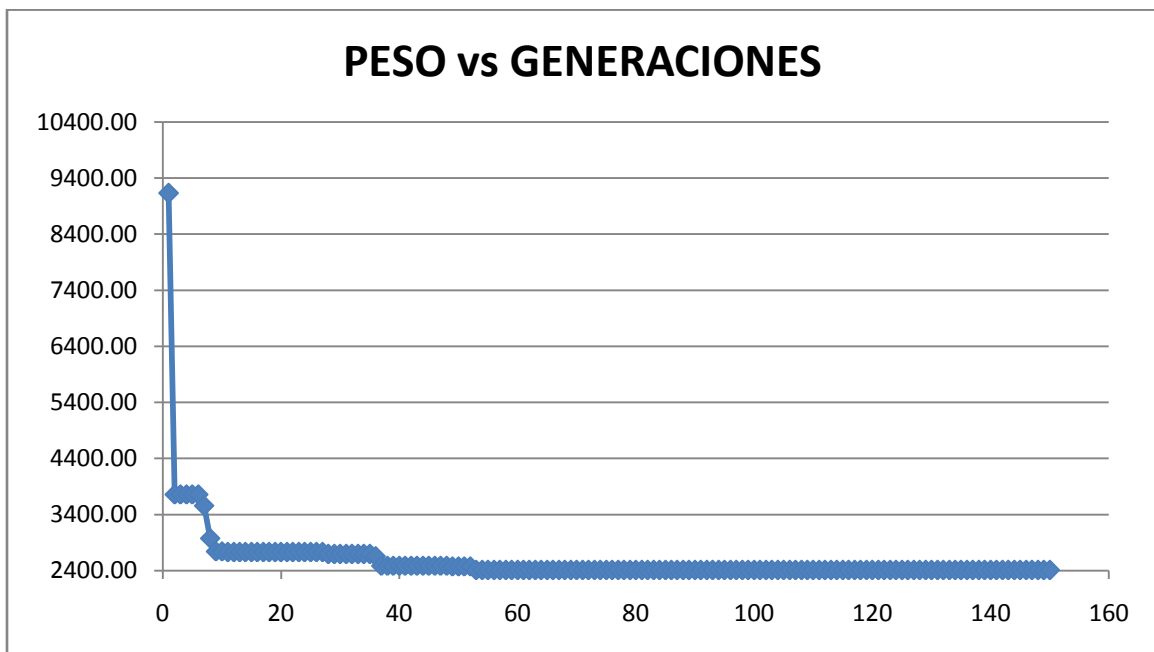
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 100 generaciones



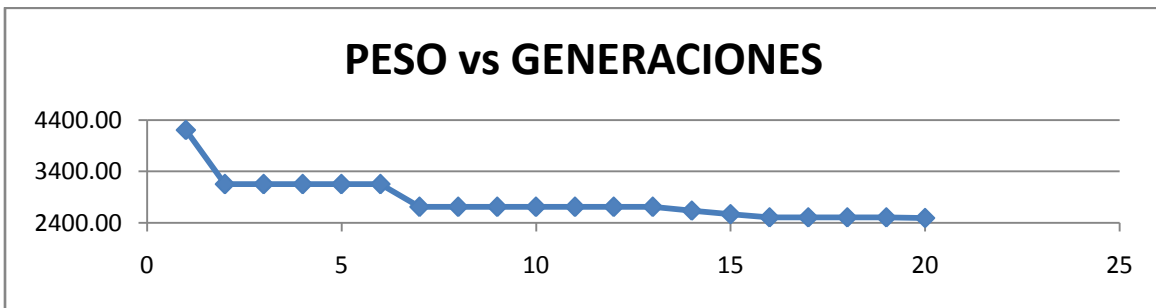
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 20 individuos con 150 generaciones



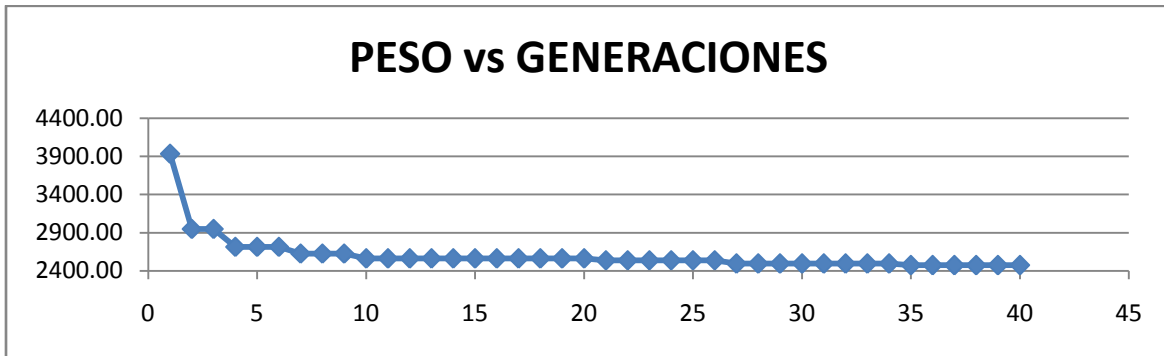
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 50 individuos con 20 generaciones



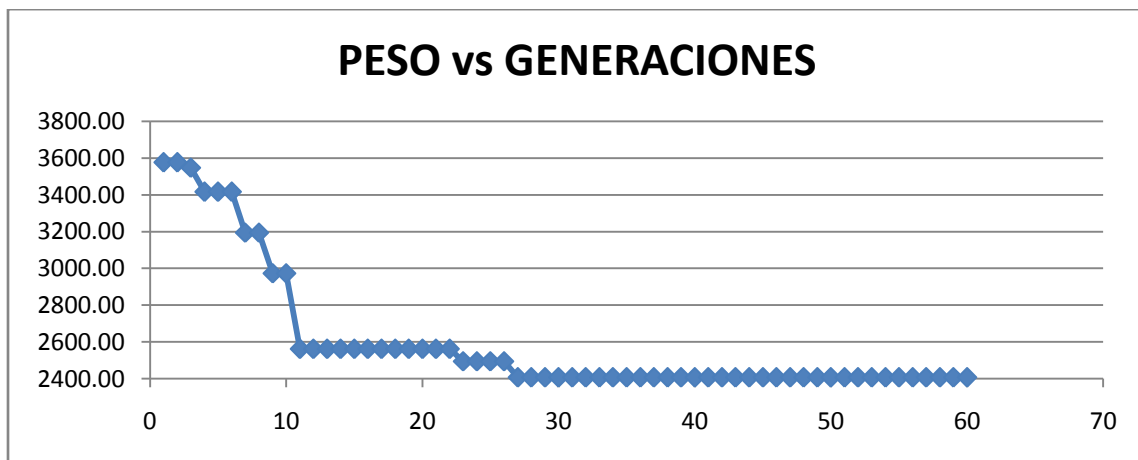
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 50 individuos con 40 generaciones



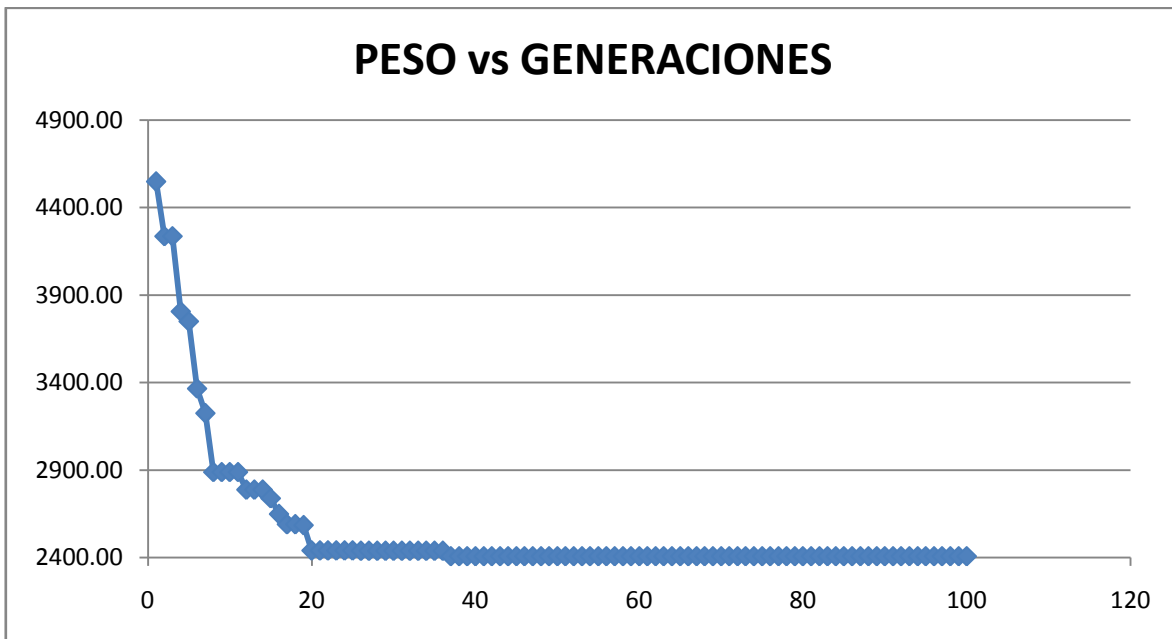
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 50 individuos con 60 generaciones



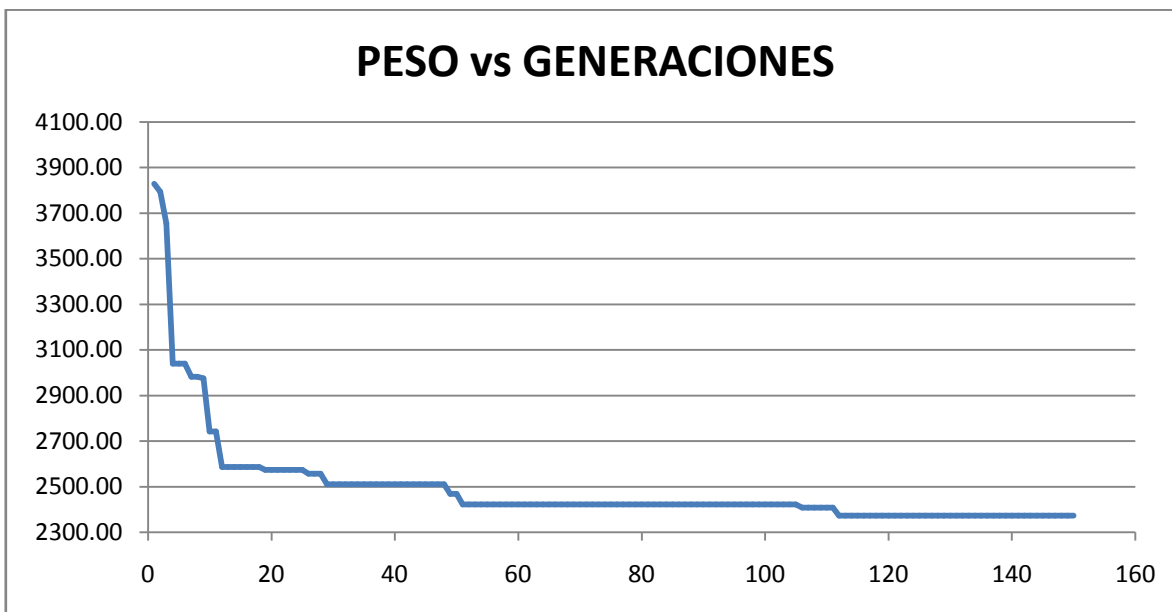
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 50 individuos con 100 generaciones



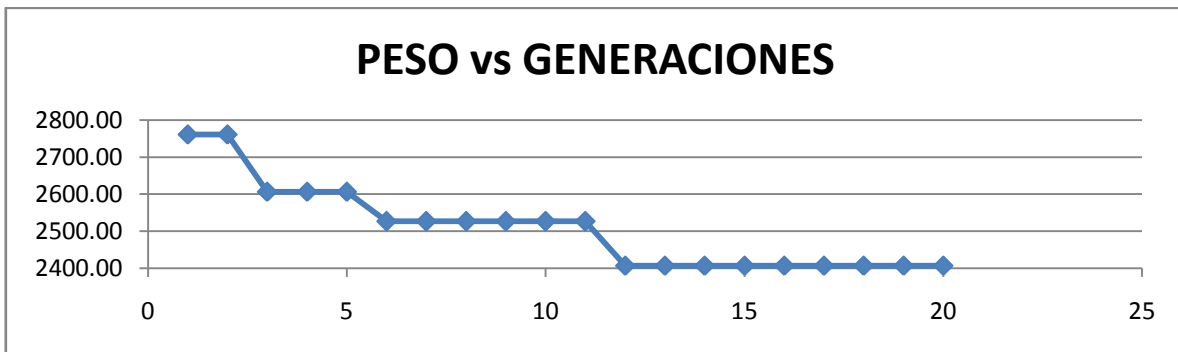
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 50 individuos con 150 generaciones



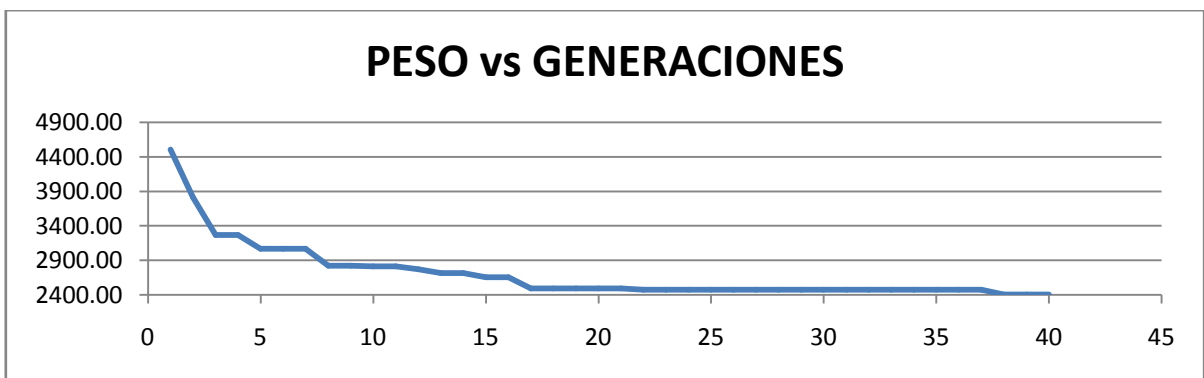
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 80 individuos con 20 generaciones



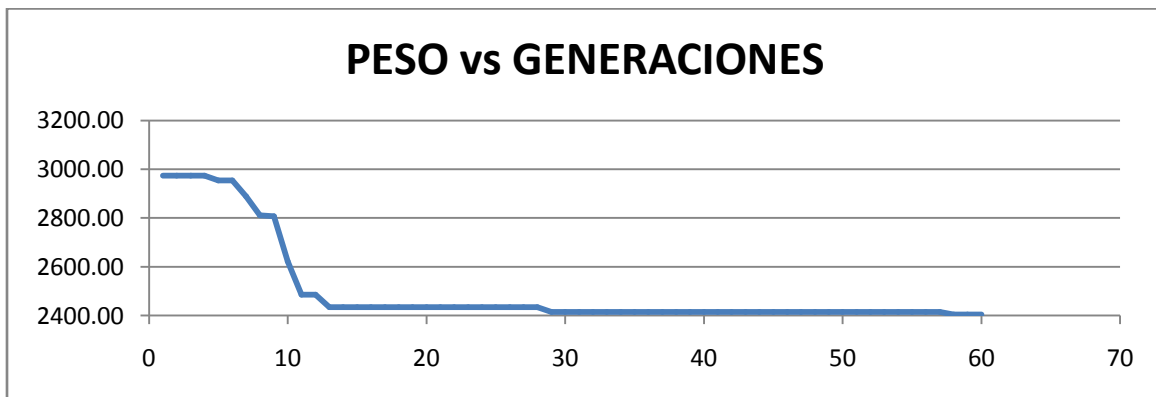
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 80 individuos con 40 generaciones



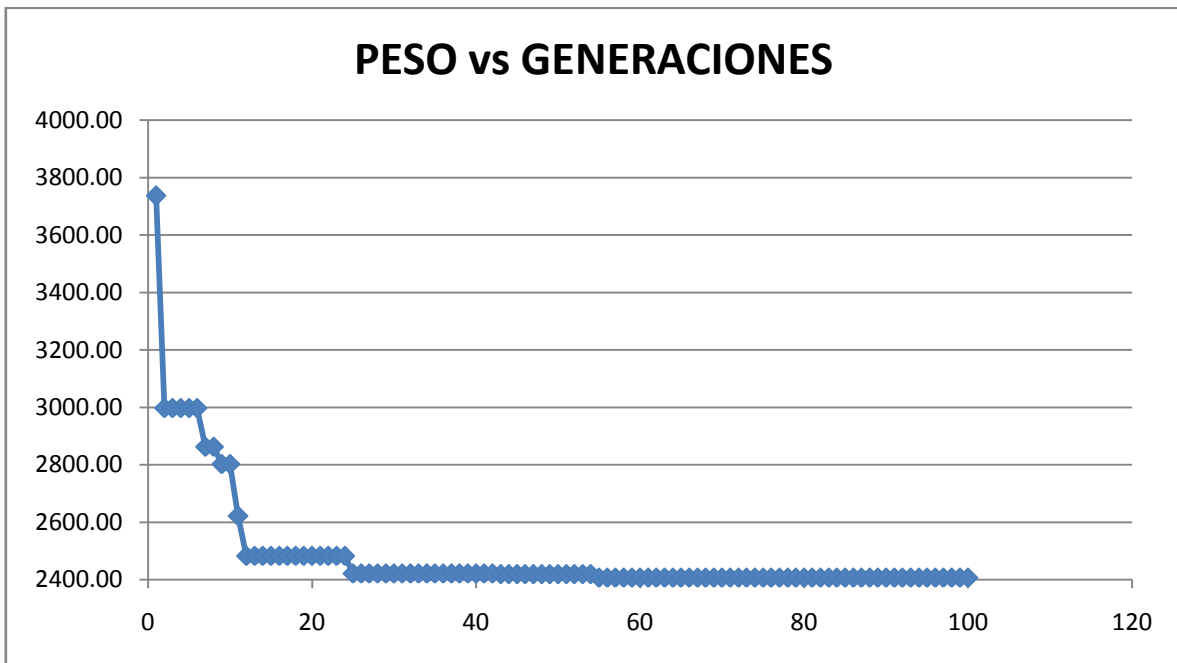
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 80 individuos con 60 generaciones



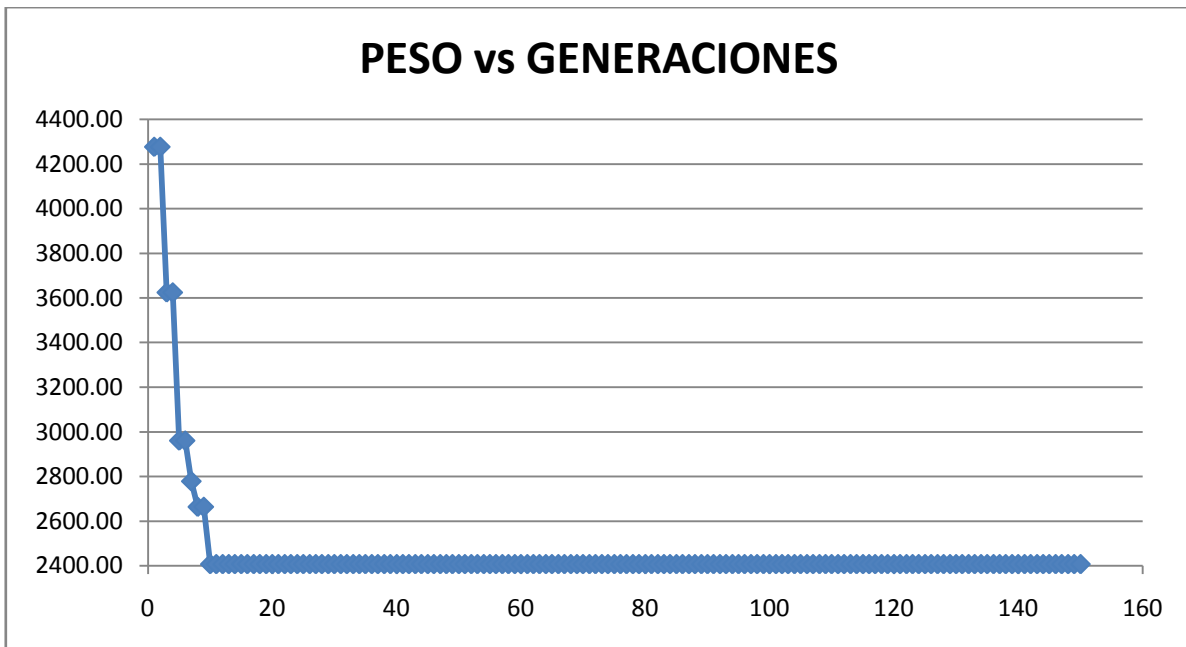
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 80 individuos con 100 generaciones



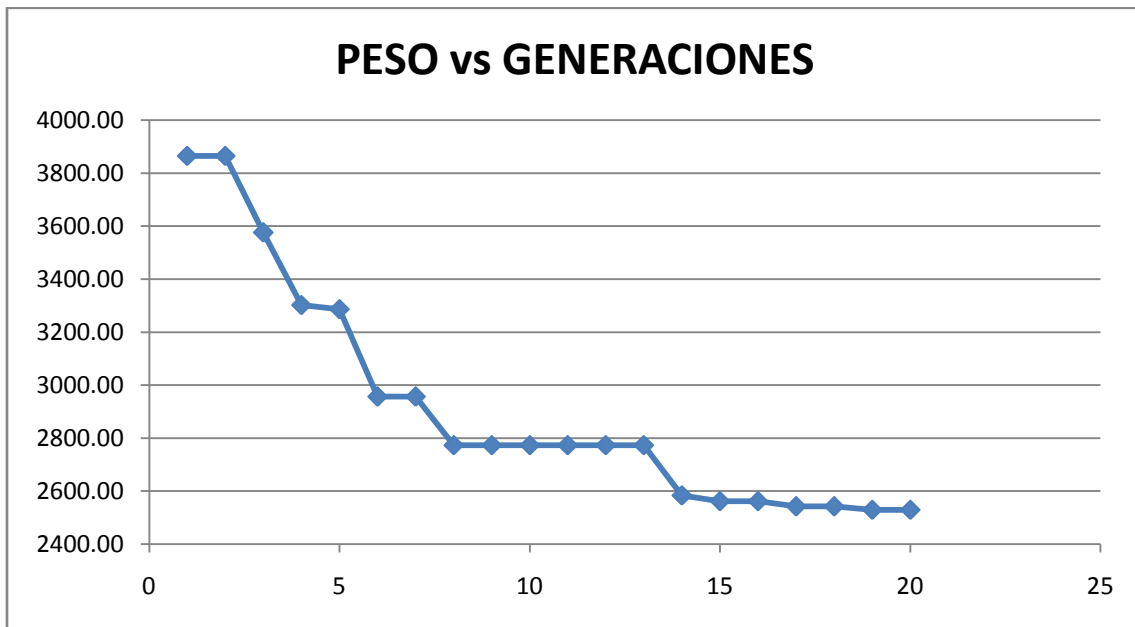
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 80 individuos con 150 generaciones



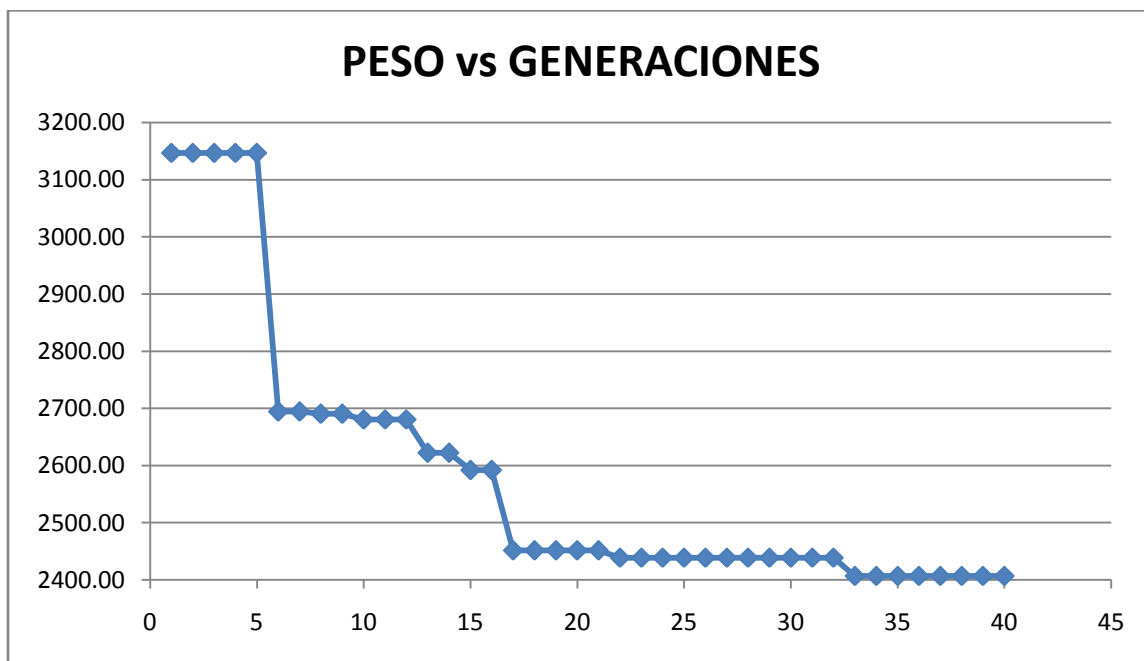
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 120 individuos con 20 generaciones



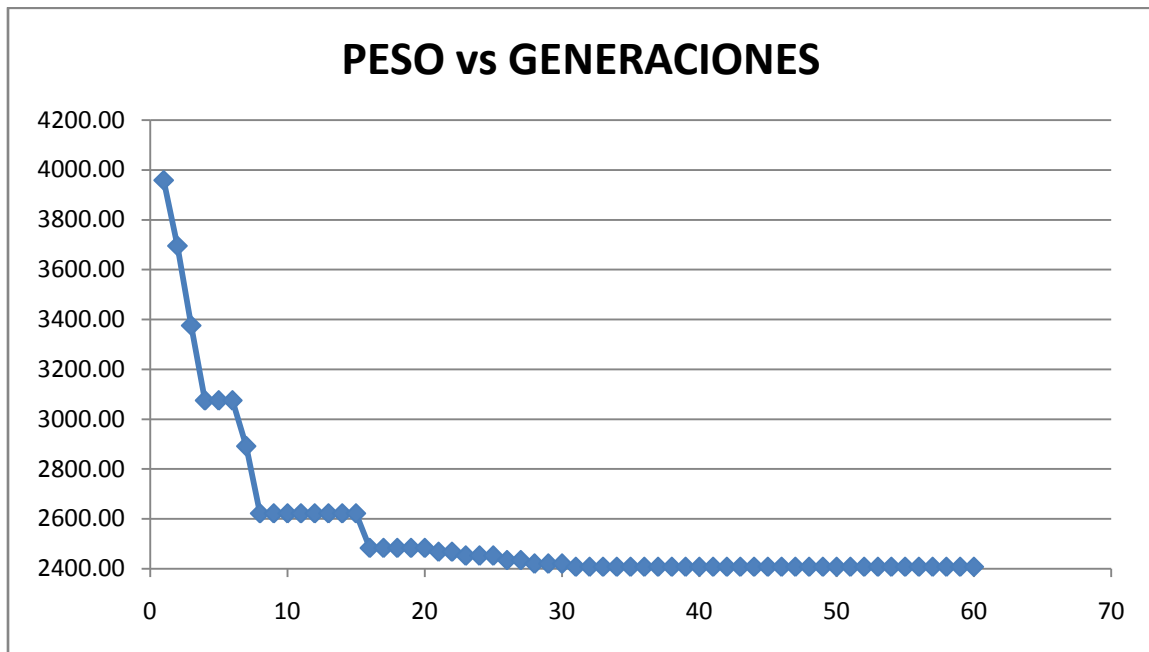
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 120 individuos con 40 generaciones



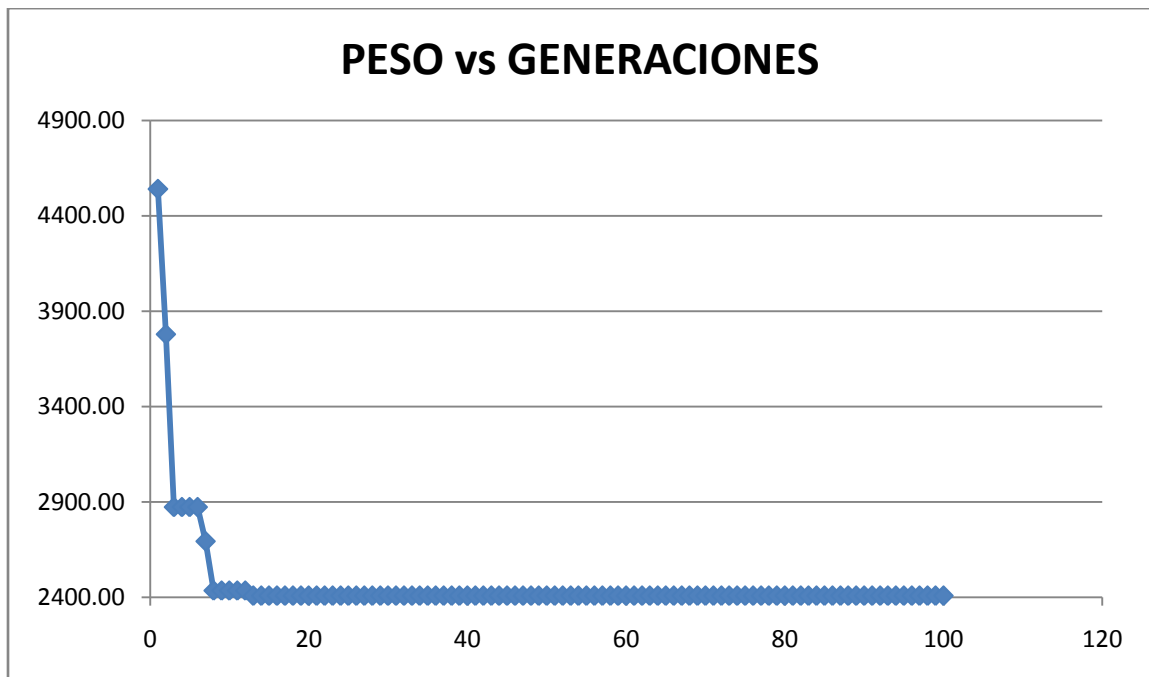
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 120 individuos con 60 generaciones



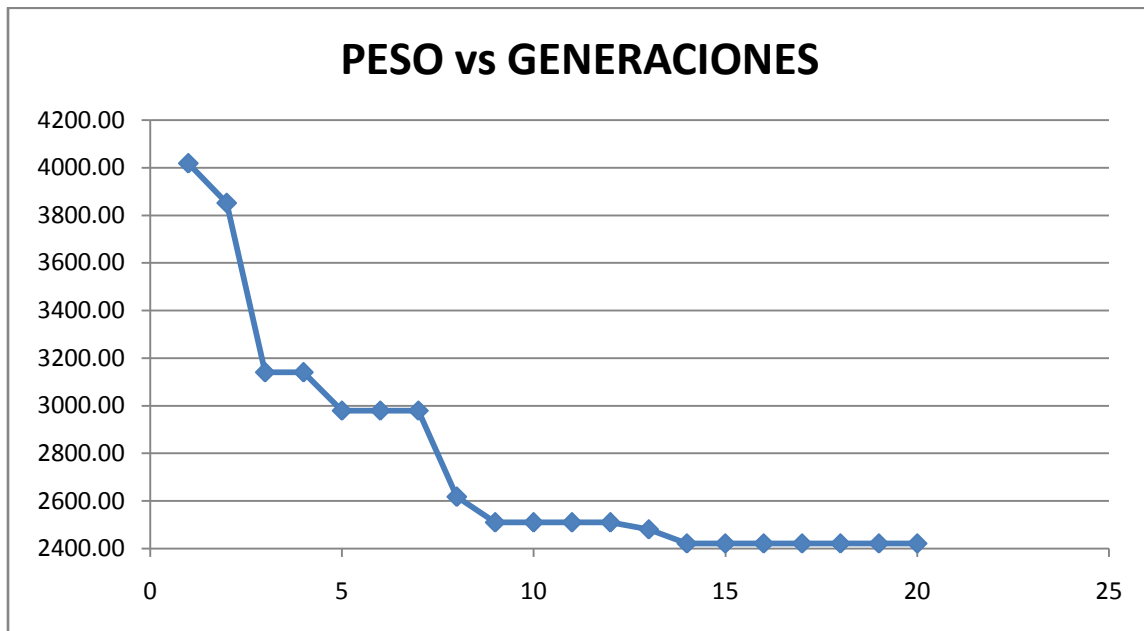
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 120 individuos con 100 generaciones



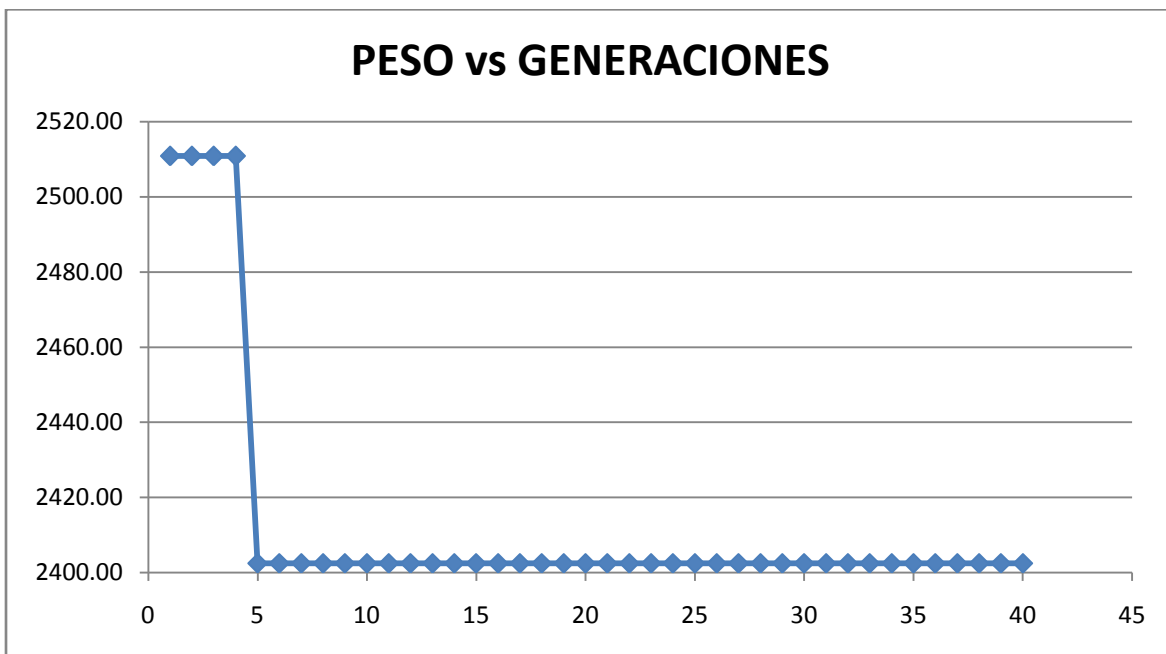
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 150 individuos con 20 generaciones



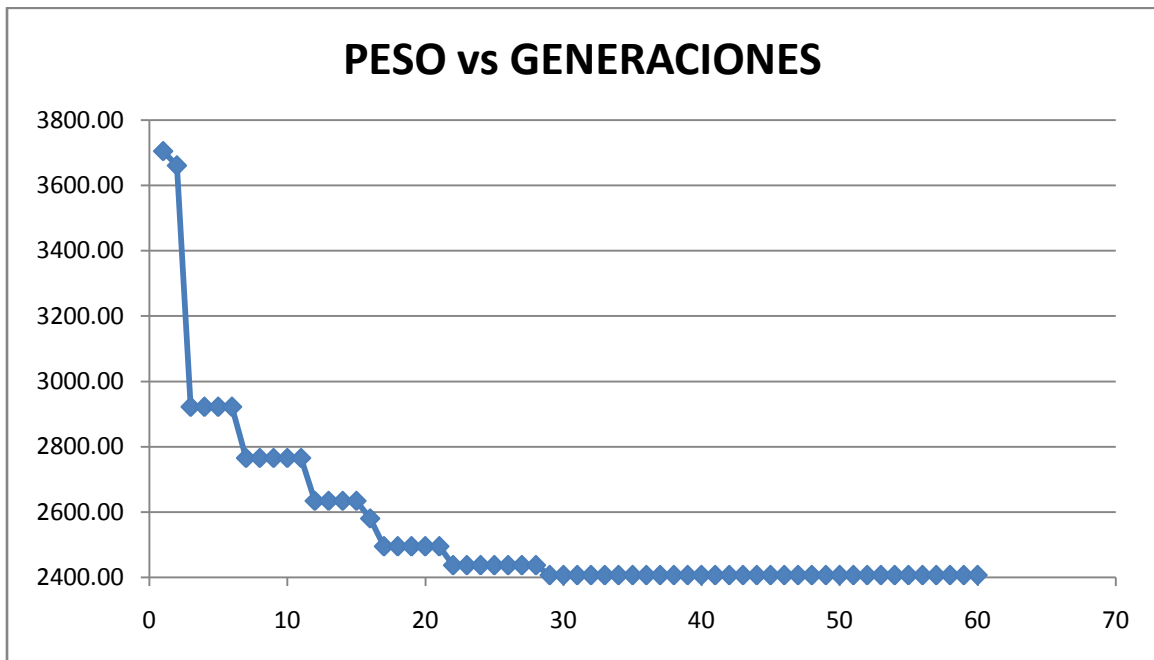
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 150 individuos con 40 generaciones



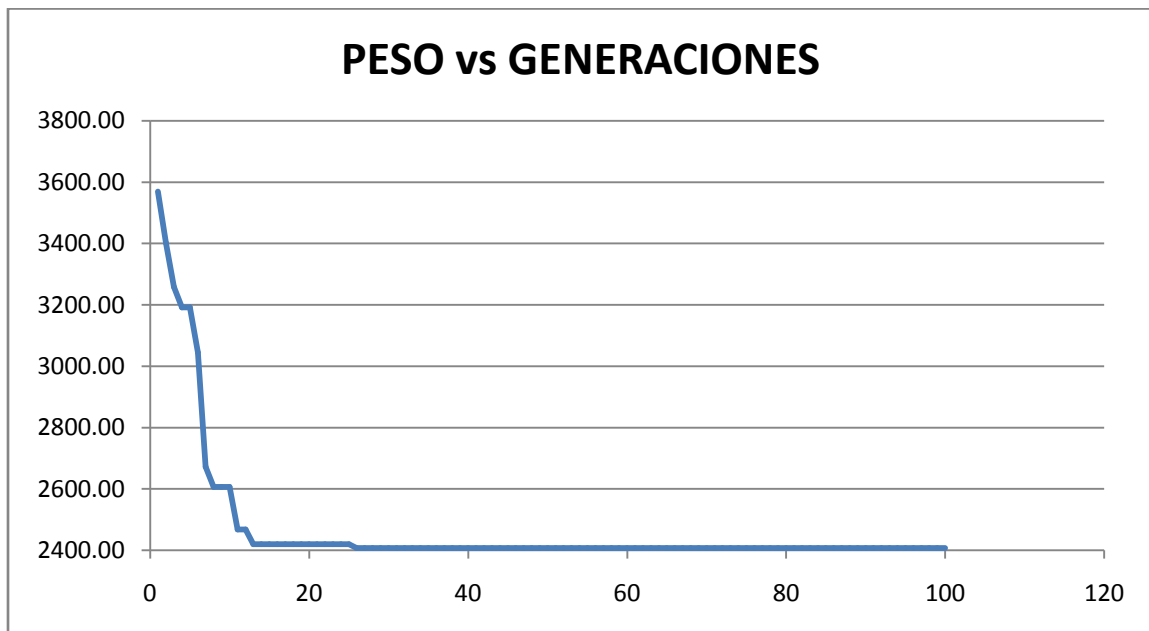
Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 150 individuos con 60 generaciones



Fuente: Elaboración propia

Figura . Variación del peso en cada generación para una población de 150 individuos con 100 generaciones



Fuente: Elaboración propia

De acuerdo con los resultados obtenidos notamos que el algoritmo hace una buena optimización a pesar de que tan solo en una población dio el resultado óptimo calculado ya con el algoritmo PSO.

Para el caso 2 notamos que el tiempo de ejecución del algoritmo para hacer la respectiva optimización se eleva considerablemente a los tiempos del caso 1, esto se debe a que el numero de variables aumenta y así aumenta el tamaño de la cadena de representación binaria, tomando más tiempo para evaluar cada individuo.

En la tabla 24 se muestra el diseño óptimo para el caso 2.

Tabla Diseño óptimo obtenido por el AG para el Caso 2

AREAS OPTIMAS				PESO
A1 [cm ²]	A2 [cm ²]	A3 [cm ²]	A4 [cm ²]	OPTIMO (kg)
4,42	4,42	6,73	7,11	2372,22

Fuente: Elaboración propia

El mejor diseño o el peso óptimo obtenido mediante el algoritmo genético para los dos casos de agrupación de miembros, es similar al peso óptimo obtenido por la optimización con el algoritmo PSO. Comprobando el buen funcionamiento de los algoritmos.

La tabla 25 muestra los resultados de la optimización hecha en artículo y la hecha en este trabajo.

Tabla Comparación del peso óptimo reportado en el artículo y el resultado óptimo obtenido en este trabajo.

PESO OPTIMO [kg]	
ARTICULO (Vedat Toğan, 18 January 2006)	3217,17
ESTE TRABAJO	3402,58

Fuente: Elaboración propia

4.4 PRESUPUESTO PARA COSTOS DEL MATERIAL

En la tabla 26 se muestran las características de los perfiles utilizados en el diseño, área, diámetro, espesor, el valor para un perfil de 6 m, y costo por kilogramo de cada perfil.

Tabla Especificaciones de los Perfiles utilizados en el diseño

AREA [cm ²]	DIAMETRO [Pulg]	ESPESOR [mm]	COSTO [\$ /6 m]	COSTO [\$ / kg]
4,42	2 1/2 "	2	73990	3554
4,51	2 "	2,5	73990	3484
6,73	3 "	2,5	110460	3480
7,11	4 "	2	117180	3500
8,72	4 "	2,5	142975	3484
10,41	4 "	3	171360	3496

Fuente: Elaboración propia

En las tablas 27, 28, 29 se muestra el peso de cada grupo de áreas, además su respectivo costo, así como también el peso total de la estructura y el costo total del material, para el diseño inicial propuesto, para el diseño óptimo obtenido con los dos algoritmos para el caso 1 y para el diseño óptimo obtenido con los dos algoritmos para el caso 2 respectivamente.

Tabla Peso y costo para el diseño inicial propuesto por los autores

GRUPO	AREA [cm ²]	PESO [kg]	COSTO
1	4,51	637,20	2'219.700
2	10,41	2557,58	8'940.590
3	7,11	671,65	2'350.773
TOTAL		3866,43	13'511.063

Fuente: Elaboración propia

Se demuestra el presupuesto para el diseño inicial siendo este de 13'511.063 de pesos.

Tabla Peso y costo para el diseño optimo para el Caso 1

GRUPO	AREA [cm ²]	PESO [kg]	COSTO
1	4,42	624,60	2'219.700
2	8,72	2141,23	7'459.622
3	6,73	636,74	2'215.962
TOTAL		3402,58	11895283

Fuente: Elaboración propia

Se demuestra el presupuesto para el diseño óptimo para el caso 1 siendo este de 11'895.283 de pesos.

Tabla Peso y costo para el diseño optimo para el Caso 2

GRUPO	AREA [cm ²]	PESO [kg]	COSTO
1	4,42	624,60	2'219.700
2	4,42	1045,80	3'716.566
3	6,73	636,74	2'215.962
4	7,11	65,07	227.757
TOTAL		2372,22	8'379.985

Fuente: Elaboración propia

Se demuestra el presupuesto para el diseño óptimo para el caso 1 siendo este de 8'379.985 de pesos.

Tabla Pesos y costos para los tres diseños, y porcentaje de reducción respecto al diseño inicial propuesto

DISEÑO		PESO [kg]	COSTO [\$]	% REDUCCION PESO	% REDUCCION COSTO
INICIAL		3866,43	13'511.063	-----	-----
OPTIMO	CASO 1	3402,58	11'895.283	12,00	11,96
	CASO 2	2372,22	8'379.985	38,65	37,98

Fuente: Elaboración propia

En la tabla 30 se observan los resultados finales obtenidos en cuanto al peso y el costo del diseño inicial y el diseño óptimo para los dos casos de agrupación de miembros, se hace una comparación en peso y en costo con el diseño inicial.

Los resultados obtenidos dan un buen balance en cuanto a la reducción del peso y del costo para los diseños obtenidos en este trabajo.

Lo anterior confirma que la agrupación de miembros propuesta en este trabajo es un muy buen diseño ya que los porcentajes de reducción de peso y costo del material están alrededor de 38%.

5. CONCLUSIONES Y RECOMENDACIONES

- En esta investigación se hizo el análisis de fuerzas estáticas y dinámicas para la estructura en estudio, así se puede decir que las fuerzas dinámicas en los miembros son insignificantes si las comparamos con las demás fuerzas axiales producidas por las otras cargas externas.
- Los dos métodos utilizados para hacer la optimización, en este caso los algoritmos genéticos (AG) y la optimización por enjambre de partículas (PSO), son productivos en cuanto a la velocidad de operación, fácil implementación en un lenguaje de programación en este caso Matlab®, buen rendimiento, resultados aceptables, así, los dos métodos se pueden adaptar sin mayores inconvenientes para trabajar con variables discretas, ya sea en números reales o en codificación con números binarios.
- Los resultados obtenidos por el algoritmo PSO son satisfactorios porque hizo una buena optimización en cualquiera de los casos de agrupación de miembros propuestos, la convergencia del algoritmo hacia el óptimo se vio reflejada en cada una de las ejecuciones.
- El algoritmo PSO programado hizo un eficiente trabajo ya que la optimización para cualquier tamaño del enjambre lleva a buenos resultados y en un tiempo relativamente corto.
- En el algoritmo PSO se nota que al aumentar el tamaño del enjambre es posible que en las primeras posiciones, generadas aleatoriamente por el

programa, se encuentre la mejor partícula para el peso óptimo de la estructura, esto se debe a que hay mas candidatos posibles para ser la mejor partícula y hace que converja mucho más rápido.

- En el algoritmo genético nunca se llegó a la mejor solución en la primera generación, por el contrario los primeros individuos generados aleatoriamente estaban muy lejos de ser los mejores, se necesitaban varias generaciones para poder hacer que convergiera.
- A mayores tamaños de población y más números de generaciones, más probabilidades de encontrar el mejor individuo en el espacio de búsqueda.
- La codificación de las variables del problema en números binarios es una gran ayuda para la optimización mediante algoritmos genéticos, ya que permite usar de manera eficiente y fácil los operadores genéticos de cruce y mutación.
- La función de aptitud o de fitness propuesta está bien diseñada, ya que los resultados obtenidos son satisfactorios. Hay que resaltar que esta función es la clave para que el AG proporcione buenos resultados, pues ésta es la que nos dice cuales son los mejores y peores individuos de la población.
- Los operadores genéticos hacen que se explore mejor el espacio de búsqueda y hacen que se encuentre de una manera más fácil y rápida el mejor individuo.

- El algoritmo genético (AG) programado presenta resultados que se asemejan a los producidos por el PSO, la convergencia y estabilidad del algoritmo se ve reflejada en los resultados obtenidos. Aunque el AG lleva más tiempo para hacer la optimización que el PSO los resultados obtenidos en cuanto a la optimización son iguales.
- El diseño inicial propuesto es un diseño aceptable y se podría presentar normalmente en este tipo de estructuras, aunque mediante los dos algoritmos PSO y AG, se comprueba que dichos perfiles escogidos no eran los mejores para el diseño, ya que se hizo una disminución del peso llevando así a reducir de forma considerable el costo del material utilizado para la construcción de la cercha.
- La propuesta de una diferente agrupación de miembros, es decir la del Caso dos nace de observar y detallar las fuerzas internas de los elementos y notar que algunos elementos soportaban fuerzas mayores a los similares en su agrupación, por esto salen de este grupo para conformar uno nuevo, y así comprobar que el diseño se hace más eficiente a medida que se analizan los resultados obtenidos por la programación ya realizada; de forma puntual, el costo del material utilizado tuvo una reducción alrededor de 38%, demostrando que es la mejor opción de diseño, lo cual es muy importante para el constructor, aunque se haya elevado el número de variables y el tiempo de ejecución de los algoritmos.
- Los algoritmos empleados en este trabajo se pueden utilizar en diferentes campos de la ingeniería civil, son fáciles de manejar y presentan resultados eficientes.

- Al comparar los resultados reportados en el artículo, podemos decir que no son iguales (aunque no varían mucho) ya que los requisitos de diseño que se utilizan allí son de la norma de Turquía, en cambio, los requisitos de diseño utilizados en este trabajo son tomados del Reglamento Colombiano de Construcción Sismo Resistente (NSR-10) en su Título F (Estructuras Metálicas).
- Los resultados obtenidos muestran la gran eficiencia de los algoritmos GA y PSO en cuanto a la optimización del peso de una cercha 3D. Se reducen los tiempos de diseño de manera considerable, ya que tomaría más tiempo en hacer diferentes diseño y escoger el mejor, además de convertirse en una labor tediosa.
- Para el algoritmo genético se recomienda que se estudien diferentes métodos para determinar la función de fitness, acotando que el utilizado en este trabajo es de resultados comprobados, pues evidencia el propósito de la optimización.
- Se recomienda en un trabajo futuro buscar la forma para que las probabilidades de cruce y de mutación se calculen dependiendo de los valores obtenidos en la función objetivo y su respectivo valor en la función de aptitud.
- Reducido el peso de una cercha se puede decir que la base en la que reposa dicha estructura tendrá un menor costo en cuanto a su diseño y construcción, ya que las fuerzas que actúan sobre la misma disminuirán y de manera considerable.

- Se abre la posibilidad de inclusión de más perfiles a los algoritmos realizados, aclarando que este trabajo se basa en perfiles comerciales, fáciles de conseguir en el mercado colombiano y para este caso son suministro de tubos Colmena, enfatizando en los perfiles tubulares únicamente para darle un control a la expansión del proyecto, pero se deja abierta la posibilidad para diseñar con otras secciones de perfiles, elucidando que se tendrían que realizar modificaciones al programa para el ámbito del diseño puesto que los controles del mismo están realizados exclusivamente para los perfiles estudiados aunque de igual forma se incluirían las condiciones adoptadas en la norma NSR10.
- Como aporte claro y específico de este proyecto, se muestra la optimización con un criterio nuevo, surge la idea de la nueva agrupación disponible para la cercha estudiada y así, con esto, mostrar que es de vital importancia tener claros los conocimientos del diseño de estructuras para llegar al punto de no sólo seguir un camino ya mostrado sino optar por proponer e implementar uno nuevo a criterio del diseñador.

BIBLIOGRAFÍA

Bathe, K.-J. (1982). *Finite element procedures in engineering analysis*. New Jersey: Prentice-Hall.

Blickle, & Thiele. (1995). *A comparison of selection schemes used in genetic algorithms*. Zurich, Switzerland: echnical Report 11, Computer Engineering and Comunnication Network Lab (TIK), Gloristrasse 35, 8092.

Cabezas Rivas, C. A. (2002). Algoritmos genéticos, una opción par la optimización de funciones. *Tecnología Química Vol. XXII, No. 2*, <http://ojs.uo.edu.cu/index.php/tq/article/viewFile/1990/1543> .

Castilho, V. C. (Febrero de 2003). *Otimização de componentes de Concreto Pré-moldado Protendidos mediante Algoritmos Genéticos*. Tesis de Doctorado, Universidade de São Paulo, São Paulo.

Celis Melo, C. E., & Millan Paramo, C. A. (2009). *Evaluación del desempeño de algoritmos de evolución parametrica para optimización*. Bucaramanga, Colombia: Tesis de grado para la obtención del título de Ingeniero Civil, Facultad de Ingenierias Fisico-Mecanicas, Universidad Industrial de Santander.

COLMENA Acero en evolución. (s.f.). Obtenido de www.tuboscolmena.com

Conte, S. D., & Boor, C. (1987). *Análisis Numerico*. Mexico D.F.: McGraw-Hill.

Cruz Chávez, M. A. (Agosto 18 del 2000). *Máximos y mínimos*. Cuernavaca Morelos, México: www.uaem.mx/posgrado/mcruz/maximominimo.PDF; UAEM Av. Universidad Chamilpa C.P.

Dorigo, M., & Di Caro, G. (1999). *The Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization*. London, UK: McGraw-Hill.

Dowland, K. (1993). *Simulated annealing*. New York: C.R. Reeves (ed.), Modern heuristic techniques for combinatorial problems.

Glover, F. y. (1993). *Tabu search*. New York: C.R. Reeves (ed.), Modern heuristic techniques for combinatorial problems.

Goldberg, D. (1989). Genetic Algorithm in search, optimization and machine learning. *Adisson-Wesley Publishing, Reading, USA* .

J. Kennedy, R. E. (December 1995). Particle swarm optimization. *Vol.4*.

J. Kennedy, R. E. (2001). Swarm intelligence.

L.J. Li, Z. H. (18 February 2009). A heuristic particle swarm optimization method for truss structures with discrete variables. *87 (2009)*.

Millonas, M. (1994). Swarms, phase transitions, and collective intelligence.

Pérez López, J. R. (Octubre de 2005). *Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo*. Santander, España: Tesis doctoral, Departamento De Ingeniería De Comunicaciones, Universidad de Cantabria.

Pukkala, T., & Kangas, J. (1993). *A heuristic optimization method for forest planning and decision making*.

Salazar Pinto, P. Y. (2009). *Algoritmo híbrido auto configurado para optimización estructural*. Bucaramanga, Colombia: Tesis de grado para la obtención del título de Ingeniero Civil, Facultad de Ingenierías Físico-Mecánicas, Universidad Industrial de Santander.

Shyue-Jian Wu, P.-T. C. (4 April 2000). Integrated discrete and configuration optimization of trusses using genetic algorithms.

Vedat Toğan, A. T. (18 January 2006). Optimization of 3d trusses with adaptive approach in genetic algorithms. *Engineering Structures 28 (2006)* , 1019–1027.

ANEXO

Este anexo está comprendido en el CD que contiene las modelaciones realizadas en el desarrollo del proyecto, el CD contiene los siguientes archivos:

- AG_3_AREAS_CERCHA_3D, el programa desarrolla el caso 1, referenciado en el trabajo, de la cercha por algoritmos genéticos.
- AG_4_AREAS_CERCHA_3D, el programa desarrolla el caso 2, referenciado en el trabajo, de la cercha por algoritmos genéticos.
- PSO_3_AREAS_CERCHA_3D, el programa desarrolla el caso 1, referenciado en el trabajo, de la cercha por optimización por enjambre de partículas.
- PSO_4_AREAS_CERCHA_3D, el programa desarrolla el caso 1, referenciado en el trabajo, de la cercha por optimización por enjambre de partículas.
- SOLUCION_CERCHA3D_ESTATICA_Y_DINAMICA, el programa analiza las cargas muertas de la estructura y su desarrollo dinámico por el método de Newmark junto con las condiciones de diseño básicas para perfiles tubulares.