

# Uso de una Técnica Multi Objetivo (Niche Pareto Genetic Algorithm) para el manejo de restricciones en un Algoritmo Genético. Resultados preliminares.<sup>1</sup>

Efrén Mezura Montes

[emezura@lania.mx](mailto:emezura@lania.mx)

Laboratorio Nacional de Informática Avanzada (LANIA)

Maestría en Inteligencia Artificial

Universidad Veracruzana.

Agosto, 2000

## Resumen

*En este reporte se explica el uso de la Técnica MultiObjetivo Niche Pareto Genetic Algorithm basada en una selección por torneo con características especiales (utilizado en Optimización Multi Objetivo). En cada selección se escogen dos individuos candidatos los cuales competirán, primero entre ellos basándose en factibilidad y si esto no es posible competirán contra un subconjunto de la población y el que resulte no dominado por ellos será el ganador. En caso de empates se selecciona aquel que viole en menor cantidad todas las restricciones.*

## 1 Introducción

El manejo de restricciones en los Algoritmos Genéticos (AGs) requiere de incorporar a él un mecanismo para elegir aquellas soluciones que orienten la búsqueda a la zona factible y de ahí a encontrar el óptimo global. Este mecanismo, aún en discusión por la comunidad de Computación Evolutiva tiene diversas vertientes. Una de ellas es el uso de funciones de penalización, donde el problema es encontrar buenos factores de penalización y ellos mismos afectan el desempeño de la búsqueda.

En este reporte se propone el uso de la selección propia del Niche Pareto Genetic Algorithm (NPGA) para el manejo de restricciones. Se realizaron modificaciones a esta manera de seleccionar individuos con el objeto de orientar de manera más rápida la búsqueda a la zona factible del espacio de búsqueda.

## 2 Selección por Factibilidad y No Dominancia basada en NPGA.

La manera en la que se adaptó NPGA para el manejo de las restricciones fue la siguiente: Para mantener el enfoque Multi Objetivo se tomaron tanto a la función objetivo como a las restricciones como objetivos aparte (no como en las funciones de penalización). Así, si se tienen  $k$  restricciones se tienen  $k+1$  objetivos. A esto lo llamamos vector de aptitudes.

Para el proceso de selección se tiene primero la opción de que si una solución es factible y la otra no, se preferirá a la factible. Por otro lado, si ambas son factibles o no factibles se procede a una selección por no dominancia.

Si una solución es no dominada por todos los elementos de un subconjunto de la población elegido aleatoriamente (es mejor en más posiciones del vector de aptitud) entonces se selecciona. Si ocurre un empate (ambos son no dominados o ambos son dominados) entonces se seleccionará aquella que viole en menor cantidad acumulada las restricciones solamente.

La comparación de no dominancia basa su presión de selección en el tamaño del subconjunto de la población elegido para la misma. Entre más grande sea, mayor presión de selección habrá. A este tamaño se le conoce como  $tdom$  [1]

Algoritmo:

---

<sup>1</sup> Reporte técnico LANIA A.C. Agosto 2000.

Seleccionar al azar candidato\_1  
 Seleccionar al azar candidato\_2  
 Seleccionar tdom individuos de la población  
 Si candidato\_1 es factible y candidato\_2 es no factible  
     Seleccionado = candidato\_1  
 Si no si candidato\_2 es factible y candidato\_2 es no factible  
     Seleccionado = candidato\_2  
 Si no  
     Compara a candidato\_1 con no\_dominancia contra los tdom individuos  
     Compara a candidato\_2 con no\_dominancia contra los tdom individuos  
     Si candidato\_1 es no dominado y candidato\_2 es dominado  
         Seleccionado = candidato\_1  
     Sino si candidato\_2 es no dominado y candidato\_1 es dominado  
         Seleccionado = candidato\_2  
 Si no  
     Si violaciónDeCandidato\_1 < ViolaciónDeCandidato\_2  
         Seleccionado = Candidato\_1  
     Si no  
         Seleccionado = Candidato\_2

### 3 Operadores Genéticos

Para este trabajo se utilizó un AG con representación real, donde en cada gen se codifica de manera directa el valor de una variable del problema, con una precisión 6 dígitos. Por ello el mapeo del genotipo al fenotipo no existe; o dicho de otra manera, es directo y no existe una función de decodificación. Una excepción se encuentra en la función de ejemplo 1, donde las dos primeras variables son múltiplos enteros de 0.0625. Así que el gen que corresponda a ese valor que en principio es un real, debe truncarse y multiplicarse por 0.0625 para obtener el valor de la variable. Sin embargo esto se presenta sólo en esas dos variables ( $x_1$  y  $x_2$ ) de ese problema

1.232233	9.458245	-9.03232	0.000021
----------	----------	----------	----------

Figura 1 Ejemplo de codificación real.

Para la cruce se utilizó Simulated Binary CrossOver en su versión para variables con límites inferior y superior. Su elección fue en base a los buenos resultados que ha mostrado en trabajos anteriores y que su capacidad de generar hijos lejos de los valores de los padres se controla por un parámetro.[2]

El procedimiento para obtener hijos  $c_1$  y  $c_2$  a partir de dos padres  $y_1$  y  $y_2$  se realiza la siguiente:

1. Generar un número random  $u$  entre 0 y 1
2. Encontrar un parámetro  $\beta q$  utilizando una distribución con probabilidad polinomial[2]:

$$\mathbf{b}_q = \begin{cases} (u\mathbf{a})^{\frac{1}{n_c+1}} & u \leq \frac{1}{\mathbf{a}} \\ \left(\frac{1}{2-u\mathbf{a}}\right)^{\frac{1}{n_c+1}} & otherwise \end{cases}$$

$$\text{Donde } \mathbf{a} = 2 - \mathbf{b}^{-(n_c+1)} \quad \text{y} \quad \mathbf{b} = 1 + \frac{2}{y_2 - y_1} \min [(y_1 - y_l), (y_u - y_2)]$$

Tenemos  $y_1$  y  $y_u$  son los límites inferior y superior de la variable correspondiente al gen a cruzar.

$n_c$  es el índice de distribución de SBX y toma un valor no negativo. Para efectos de este reporte se utilizó  $n_c=1$

3. Los hijos se calculan de la siguiente manera:

$$c_1 = 0.5[(y_1 + y_2) - \mathbf{b}|y_2 - y_1|]$$

$$c_2 = 0.5[(y_1 + y_2) + \mathbf{b}|y_2 - y_1|]$$

Su complemento seleccionado fue la Parameter Based Mutation, donde la perturbación del operador está en función del número de generación.[2]. Su funcionamiento se describe a continuación:

1. Generar un número random  $u$  entre 0 y 1
2. Calcular el parámetro  $\delta_q$  de la siguiente manera:

$$\mathbf{d}_q = \begin{cases} \left[2u + (1 - 2u)(1 - \mathbf{d})^{n_m+1}\right]^{\frac{1}{n_m+1}} - 1 & u \leq 0.5 \\ 1 - \left(2(1 - u) + 2(u - 0.5)(1 - \mathbf{d})^{n_m+1}\right)^{\frac{1}{n_m+1}} & otherwise \end{cases}$$

Donde  $\mathbf{d} = \min [(y - y_l), (y_u - y)] / (y_u - y_l)$ . El parámetro  $n_m$  es el índice de distribución par la mutación y toma un valor no negativo. Para efectos de este reporte  $n_m=100+t$  donde  $t$  es el número de generación actual.

3. Calcular a la posición mutada:  
 $c = y + \mathbf{d}_q (y_u - y_l)$

## 4 Ejemplos

Para probar este enfoque se utilizaron tres funciones que representan problemas de diseño en Ingeniería Civil.[3] Los resultados se comparan con soluciones encontradas con otras técnicas.

### 4.1 Ejemplo 1: Pressure Vessel Design.

Un recipiente cilíndrico con tapas hemisféricas donde se quiere minimizar el costo del material, forma y soldadura. Se tienen cuatro variables de diseño:  $T_s$  (espesor del cuerpo),  $T_h$  (espesor de la tapa),  $R$  (radio del recipiente) y  $L$  (longitud del recipiente)

Minimizar:

$$F(X) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Sujeta a:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\mathbf{p}x_3^2x_4 - \frac{4}{3}\mathbf{p}x_3^3 + 1,296,000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

#### 4.2 Ejemplo 2 Welded Beam Design

Una viga soldada se diseña con mínimo costo sujeta a restricciones de esfuerzo cortante, carga en la barra, deflexión final y restricciones de tamaño. Existen cuatro variables de diseño  $h(x_1)$ ,  $l(x_2)$ ,  $t(x_3)$ ,  $b(x_4)$ .

Minimizar:

$$F(X) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

Sujeta a:

$$g_1(X) = \mathbf{t}(X) - \mathbf{t}_{\max} \leq 0$$

$$g_2(X) = \mathbf{s}(X) - \mathbf{s}_{\max} \leq 0$$

$$g_3(X) = x_1 - x_4 \leq 0$$

$$g_4(X) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$$

$$g_5(X) = 0.125 - x_1 \leq 0$$

$$g_6(X) = \mathbf{d}(X) - \mathbf{d}_{\max} \leq 0$$

$$g_7(X) = P - P_c(X) \leq 0$$

donde:

$$\mathbf{t}(X) = \sqrt{(\mathbf{t}'^2) - 2\mathbf{t}'\mathbf{t}'' \frac{x_2}{2R} + (\mathbf{t}''^2)}$$

$$\mathbf{t}'(X) = \frac{P}{\sqrt{2}x_1x_2}$$

$$\mathbf{t}''(X) = \frac{MR}{J}$$

$$M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[ \frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\}$$

$$\mathbf{s}(X) = \frac{6PL}{x_4x_3^2}, \mathbf{d}(X) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(X) = \frac{4.013E \sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$P=6000 \text{ lb}, \quad L=14 \text{ in}, \quad E=30 \times 10^6 \text{ psi}, \quad G=12 \times 10^6 \text{ psi}$$

$$\mathbf{t}_{\max} = 13,600 \text{ psi}, \quad \mathbf{s}_{\max} = 30,000 \text{ psi}, \quad \mathbf{d}_{\max} = 0.25 \text{ in}$$

### 4.3 Ejemplo 3 Tension/Compression Spring Design

Consiste en minimizar el peso de un resorte de tensión/compresión sujeto a restricciones de mínima deflección, tensión de corte, frecuencia de ondulamiento, límites de diámetro exterior y en las variables de diseño. Existen cuatro variables: Diámetro del resorte (D), diámetro del cable(d) y el número de vueltas del resorte(N).

Minimizar:

$$F(X) = (N + 2)Dd^2$$

Sujeta a:

$$g_1(X) = 1 - \frac{D^3 N}{71875d^4} \leq 0$$

$$g_2(X) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0$$

$$g_3(X) = 1 - \frac{140.45d}{D^4 N} \leq 0$$

$$g_4(X) = \frac{D + d}{1.5} - 1 \leq 0$$

## 6 Comparación de Resultados

Los parámetros utilizados en los tres problemas para el Ag son:

Porcentaje de Cruza	=	0.9
Porcentaje de Mutación	=	0.9
Población	=	100 individuos
Número de generaciones	=	500
Tamaño del torneo	=	10

### 6.1 Ejemplo 1 Pressure Vessel Design

Los resultados obtenidos se comparan con los mostrados por Coello[3]. Los rangos de las variables se acotaron de la siguiente manera:  $1 \leq x_1 \leq 99$ ,  $1 \leq x_2 \leq 99$ ,  $10.000000 \leq x_3 \leq 200.000000$ ,  $10.000000 \leq x_4 \leq 200.000000$ .

La media de aptitud de las 20 corridas realizadas fue  $f(X)=192010.984653$  con una desviación estándar de 134279.0977223. La peor solución fue  $f(X)=469127.638639$ . La solución en la mediana fue  $f(X)=158764.420811$  ( $x_1=37*0.0625=2.315798$ ,  $x_2=84*0.0625=5.283844$ ,  $x_3=112.484895$ ,  $x_4=155.881350$ ).

Aquí sólo una de veinte soluciones fue no factibles

Variables de Diseño	Mejor Solución Encontrada				
	Este reporte	Coello[3]	GeneAS[4]	Kannan[5]	Sandgren[6]
$x_1(T_s)$	1.548142	0.8125	0.9375	1.125	1.125
$x_2(T_h)$	2.360910	0.4375	0.5000	0.625	0.625
$x_3(R)$	57.129872	40.3239	48.3290	58.291	47.700
$x_4(L)$	155.521163	200.0000	112.6790	43.690	117.701
$g_1(X)$	-0.445536	-0.034324	-0.004750	0.000016	-0.204390
$g_2(X)$	-1.815891	-0.052847	-0.038941	-0.068904	-0.169942
$g_3(X)$	-1079706.500007	-307.4333	-3652.876838	-21.220104	54.226012
$g_4(X)$	-84.478837	-40.000000	-127.321000	-96.310000	-122.299000
<b>f(X)</b>	<b>26159.236607</b>	<b>6288.7445</b>	<b>6410.3811</b>	<b>7198.0428</b>	<b>8129.1036</b>

Tabla 1 Comparación de Resultados para el primer ejemplo (Pressure Vessel Design)

### 6.2 Ejemplo 2 Welded Beam Design

Los resultados obtenidos se comparan con los mostrados por Coello[3]. Los rangos de las variables se acotaron de la siguiente manera:  $0.100000 \leq x_1 \leq 2.000000$ ,  $0.100000 \leq x_2 \leq 10.000000$ ,  $0.100000 \leq x_3 \leq 10.000000$ ,  $0.100000 \leq x_4 \leq 2.000000$ .

La media de aptitud de las 20 corridas realizadas fue  $f(X)=7.954134$  con una desviación estándar de 5.510053. La peor solución fue  $f(X)=28.832552$ . La solución en la mediana fue  $f(X)=5.921578$  ( $x_1=0.406536$ ,  $x_2=6.555872$ ,  $x_3=4.469915$ ,  $x_4=1.068801$ ).

Aquí sólo seis de veinte soluciones fueron no factibles.

Variables de Diseño	Mejor Solución Encontrada				
	Este reporte	Coello[3]	Deb[7]	Siddal[8]	Ragsdell[9]
$x_1(h)$	0.267106	0.2088	0.2489	0.2444	0.2455
$x_2(l)$	4.899696	3.4205	6.1730	6.2189	6.1960
$x_3(t)$	5.456669	8.9975	8.1789	8.2915	8.2730
$x_4(b)$	0.783389	0.2100	0.2533	0.2444	0.2455
$g_1(X)$	-1384.510360	-0.337812	-5758.603777	-5743.502027	-5743.826517
$g_2(X)$	-8392.829902	-353.902604	-255.576901	-4.015209	-4.715097
$g_3(X)$	-0.516283	-0.001120	-0.004400	0.000000	0.000000
$g_4(X)$	-1.105702	-3.411865	-2.982866	-3.022561	-3.020289
$g_5(X)$	-0.142106	-0.08380	-0.123900	-0.119400	-0.120500
$g_6(X)$	-0.232753	-0.235649	-0.234160	-0.234243	-0.234208
$g_7(X)$	-221184.661318	-363.232380	-4465.270928	-3490.469418	-3604.275002
<b>f(X)</b>	<b>4.273002</b>	<b>1.74830941</b>	<b>2.43311600</b>	<b>2.38154338</b>	<b>2.38593732</b>

Tabla 2 Comparación de Resultados para el segundo ejemplo (Welded Beam Design)

### 6.3 Ejemplo 3 Tension/Compression Spring Design

Los resultados obtenidos se comparan con los mostrados por Coello[3]. Los rangos de las variables se acotaron de la siguiente manera:  $0.050000 \leq x_1 \leq 2.000000$ ,  $0.250000 \leq x_2 \leq 1.300000$ ,  $2.000000 \leq x_3 \leq 15.000000$ .

La media de aptitud de las 20 corridas realizadas fue  $f(X)=0.275591$  con una desviación estándar de 0.460607. La peor solución fue  $f(X)=1.875150$ . La solución en la mediana fue  $f(X)=0.043297$  ( $x_1=0.064292$ ,  $x_2=0.806062$ ,  $x_3=10.994923$ ).

Cabe señalar que las soluciones fueron , exceptuando dos, todas no factibles.

Variables de Diseño	Mejor Solución Encontrada			
	Este reporte	Coello[3]	Arora[10]	Belegundu[11]
$x_1(d)$	0.050000	0.051480	0.053396	0.050000
$x_2(D)$	0.332533	0.351661	0.399180	0.315900
$x_3(N)$	4.744874	11.632201	9.185400	14.25000
$g_1(X)$	0.611606	-0.002080	0.000019	-0.000014
$g_2(X)$	0.037519	-0.000110	-0.000018	-0.003782
$g_3(X)$	-12.384324	-4.026318	-4.123832	-3.938302
$g_4(X)$	-0.744978	-0.731239	-0.698283	-0.756067
<b>f(X)</b>	<b>0.005607</b>	<b>0.0127047834</b>	<b>0.0127302737</b>	<b>0.0128334375</b>

Tabla 3 Comparación de Resultados para el tercer ejemplo (Tension/Compression Spring Design)

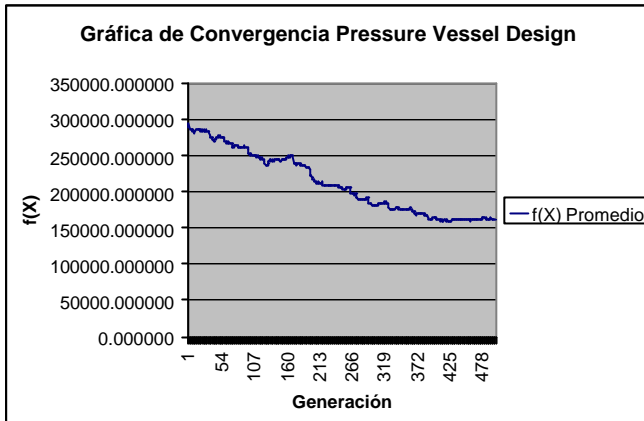


Figura 2 Gráfica de Convergencia para el ejemplo 1

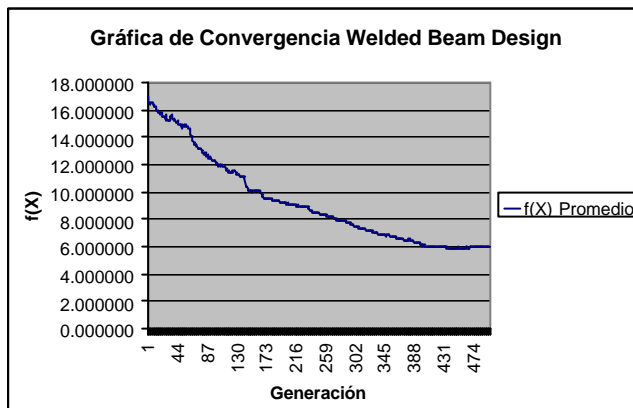


Figura 3 Gráfica de Convergencia para el ejemplo 2

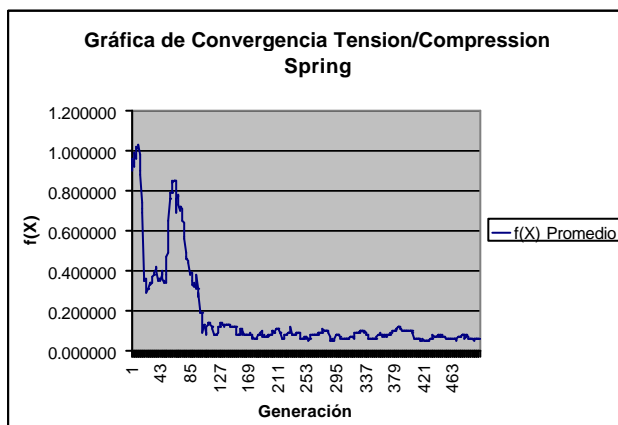


Figura 4 Gráfica de Convergencia para el ejemplo 3

Con respecto a la convergencia de los tres ejemplos puede observarse en las figuras 2, 3 y 4 que la función objetivo fue minimizada de manera uniforme, a excepción del ejemplo 3 donde en un momento del proceso la función objetivo aumentó sobremanera y luego disminuyó de la misma forma, sin poder disminuir más.

## 7 Discusión

Como puede observarse en los resultados, la técnica propuesta no mejoró, en ninguno de los tres ejemplos, las soluciones reportadas anteriormente. Parece que la presión de selección es demasiado fuerte y la convergencia prematura se presentó en muchos de los casos. Principalmente se presentaron estos comportamientos:

- 1) Si se encuentra una solución factible, la búsqueda se guía hacia ella y se tiene convergencia prematura.
- 2) No se encuentra una solución factible.

Se llevó una contabilidad de la cantidad de selecciones realizadas durante el proceso evolutivo. Al final se tiene el total de selecciones por factibilidad y el total de selecciones por no dominancia.

- 1) En las primeras generaciones al existir pocas o ninguna solución factible dominan las selecciones por no dominancia.
- 2) En etapas avanzadas del proceso, las selecciones por factibilidad aumentan, pero no de manera significativa.

Para los tres ejemplos la selección por no dominancia fue, por mucho la más utilizada. Esto indica que el número de individuos factibles por generación no fue significativo.

La frecuencia de los operadores genéticos fue propuesta para que el factor disruptivo estuvieran presente en todo momento de la búsqueda; sin embargo la convergencia prematura se presentó.

Se hicieron pruebas variando el parámetro de  $t_{dom}$  para disminuir la presión de selección. Sin embargo el resultado observado fue que la mejor aptitud global variaba más de generación en generación; es decir, si en la generación  $t$  la aptitud mejor global es  $m$ , en la generación  $t+1$  ese valor es  $m \pm n$  donde  $n$  aumenta conforme  $t_{dom}$  disminuye.

Con respecto a la factibilidad de las soluciones, para el ejemplo 1 y 2 más del 75 % de las corridas proporcionaron soluciones factibles. Por otro lado, para el ejemplo 3 sólo el 10% de las soluciones fueron factibles.

Comparando la cantidad de violación de las restricciones se observó que la búsqueda se dirigió con firmeza hacia la zona factible del espacio de búsqueda. Sin embargo, no logró satisfacer el total de las restricciones.

Con respecto a los parámetros de inicio se hicieron pruebas con porcentajes de cruza y mutación más pequeños. Esto favoreció la pérdida de diversidad en etapas tempranas de la simulación. El incrementar tanto el tamaño de la población como el número de generaciones no mostraron ejercer mejoría en los resultados.

El tiempo computacional de este enfoque es muy poco ya que se elimina el uso de nichos (en NPGA original) gracias a la opción de selección por factibilidad y al desempate por cantidad de violación de restricciones de un individuo.

Con respecto a las gráficas de convergencia no se observan en las medias de aptitud las diferencias que sí existen entre los mejores individuos por generación. Esto pudiera hacer pensar que la convergencia no es prematura, pero en varias corridas ejemplo, esto sí sucedió.

Otra causa es que para las corridas ejemplo se utilizó un tamaño de torneo de 10, lo que, según los resultados, genera una variación muy pequeña entre mejores aptitudes de las mejores soluciones globales observadas generación por generación.

Un aspecto importante de estas gráficas es que se observa que después de la generación 400 la solución ya no parece mejorar, lo que apoya la idea empírica de que el agregar más generaciones ya no mejora la aptitud de la solución.

## 8 Conclusiones

En este reporte se explica una adaptación del NPGA, originalmente creado para Optimización Multi Objetivo, para el manejo de restricciones en un Algoritmo Genético.

La técnica mostró un funcionamiento consistentemente bueno, a pesar de no mejorar soluciones publicadas previamente. Se observó un buen direccionamiento de la búsqueda hacia la zona factible y una exploración aceptable del espacio de búsqueda gracias a los operadores genéticos seleccionados.

El tiempo de ejecución de las diversas pruebas realizadas fue bastante pequeño gracias a la eliminación del uso de nichos del NPGA original y agregando la opción de selección por factibilidad y por cantidad de violación de restricciones.

## 9 Trabajo Futuro

Las áreas inmediatas de trabajo son el evitar la convergencia prematura del AG y aumentar la cantidad de soluciones factibles para problemas con mayor grado de complejidad.

Otra vertiente de investigación es el uso de otras técnicas propias de Optimización Multi Objetivo como MOGA, VEGA y NSGA para el manejo de restricciones en un AG.

## Referencias

- [1] Jeffrey Horn and Nicholas Nafpliotis. *Multiobjective Optimization Using The Niche Pareto Genetic Algorithm*, Proceedings of the First IEEE Conference of Evolutionary Computation, Department of Computer Science University of Illinois, July 1993
- [2] Kalyanmoy Deb and Samir Agrawal. *A Niche-Penalty Approach for Constraint Handling in Genetic Algorithms*. Indian Institute of Technology Kanpur, PIN 208 016, India
- [3] Carlos A. Coello Coello. *Use of a SelfAdaptive Penalty Approach for Engineering Optimization Problems*. Laboratorio Nacional de Informática Avanzada, Xalapa, Ver. México, Ago 6 2000.
- [4] Kalyanmoy Deb. GeneAS: A Robust Optimal Design Technique for Mechanical Component Design. In Dipankar Dasgupta and Zbigniew Michalewicz, editors, *Evolutionary Algorithms in Engineering Applications*, pages 497-514. Springer-verlag, Berlin 1997.
- [5] B. K. Kannan and S. N. Kramer. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and its Applications to Mechanical Design. *Journal of Mechanical Design. Transactions of the ASME*,116:318-320,1994
- [6] E. Sandgren. Nonlinear integer and discrete programming in mechanical design. In *Proceedings of the ASME Design Technology Conference*, pages 95-105, Kissimmee, Florida 1998.
- [7] Kalyanmoy Deb. Optimal Design of a Welded Beam via Genetic Algorithms *AIAA Journal*, 29(11):2013-2015, November, 1991.
- [8] James N. Siddall. *Analytical Design-Making in Engineering Design*. Prentice-Hall, 1972.
- [9] K. M. Ragsdell and D. T. Phillips. Optimal Design of a class of Welded Structures Using Geometric Programming. *ASME Journal of Engineering for Industries*,98(3):1021-1025,1976. Series B.
- [10] Jasbir S. Arora. *Introduction to Optimum Design*. McGraw-Hill , New York, 1989
- [11] Ashok Dhondu Belegundu. *A Study of Mathematical Programming Methods for Structural Optimization*. Dept. of Civil and Environmental Engineering, University of Iowa, Iowa, Iowa, 1982.