Tarea 3 Optimización de funciones con restricciones

Eduardo Tapia Romero

11 de septiembre de 2020

Índice

1. Introducción

Esta tarea tiene como objetivo el desarrollo y prueba de un algoritmo de evolución diferencial al que se le ha modificado el método de selección para que elija un cierto porcentaje de individuos de elite para generar la nueva población.

Para ello se propone utilizar el algoritmo para las funciones:

- Funcion no linear de Himmelblau
- Minimizacion del peso del peso de un resorte de Tensión compresión
- Diseño de un recipiente a presión.
- Diseño de una viga soldada.

2. Algoritmo de Evolución diferencial

El algoritmo de evolución diferencial que se va a utilizar tiene una ligera modificación al convencional, ya que al seleccionar los padres, se ordenan y se selecciona únicamente de entre los mejores de la población y no de forma aleatoria como se haría normalmente:

Algorithm 1 Algoritmo Evolución diferencial

for $i = 0; iter_{max}$ **do**
  for $j = 1; pob$ **do**
    ordenar población
    seleccionar p_1, p_2, p_3 dentro del mejor 20 % de la población
    if $rand() > CR$ **then**
      $hijo[i] = poblador[i]$
    else
      $hijo[i] = cruzar(p_1, p_2, p_3)$
    end if
    $evaluar(población)$
  end for
end for=0

Podemos observar que el algoritmo es bastante sencillo, sin embargo es necesario mencionar la regla con la que se cruzan los pobladores

$$hijo[i] = pob_{p_1} + F[(pob_{p_2} - pob_{p_3}) + (pob_{p_{best}} - pob_{p_1})] \tag{1}$$

Esto genera que en general la población tenga cierta direccionalidad a converger cerca del mejor resultado encontrado hasta el momento, sin embargo, es importante tener en cuenta que esto reduce rápidamente la diversidad, lo que puede generar que el algoritmo converja rápidamente a un mínimo local.

2.1. Manejo de las restricciones y condiciones de optimalidad.

Dado que estos problemas son problemas con restricciones, es necesario implementar una metodología que permita considerar tanto el fitness de la función per-se $f(x)$ como su factibilidad $\phi(x)$ es decir, cuantas restricciones falla en cumplir o no y por cuanto, de esta manera dando la posibilidad de encontrar soluciones factibles, aun en el caso de que las soluciones iniciales que se tengan en el algoritmo sean no factibles, para ello para cada individuo se asígna un nuevo valor de fitness mas completo:

$$fitness =< \phi(x), f(x) > \tag{2}$$

Donde fitness es una tupla de valores tales que ϕ representa las violaciones que se incurren para una solución dada, y $f(x)$ es el valor de fitness de la función en la solución respectiva. El acomodar el fitness de esta forma, permite que los individuos puedan ser comparados a partir de ambos valores, dando prioridad lexicográfica al valor de *phi* y en caso de que ambos tengan el mismo valor de violaciones, entonces se procede a comparar el valor de fintess, generando así que se converja a soluciones cada vez con menor restricciones violadas para así posteriormente encontrar preponderante mente soluciones factibles que poco a poco vayan convergiendo a algún mínimo local .

El valor de ϕ se calcula de la siguiente forma

$$\phi(x^*) = \sum_i^p |g_i(x^*)| + \sum_i^q |h_i(x^*)|, \forall g_i(x) > 0, h_i(x) \neq 0 \tag{3}$$

3. Funciones a optimizar

En esta tarea se optimizaran 4 funciones distintas, las cuales tienen sus propias restricciones, se presentan a continuación:

3.1. Himmelblau non linear función

La función no linear de himmelblau es una función que se utiliza para optimizar funciones con restricciones, consiste en minimizar:

$$f(x) = 5{,}3578547x_3^2 + 0{,}8356891x_1x_5 + 37{,}293239x_1 + 40792{,}141 \tag{4}$$

$$g_1(x) = 85{,}334407 + 0{,}0056858x_2x_5 - 0{,}00026x_1x_4 - 0{,}0022053x_3x_5 - 0{,}0022053x_3x_5 \tag{5}$$

$$g_2(x) = 80{,}51249 + 0{,}0071317x_2x_5 + 0{,}0029955x_1x_2 + 0{,}0021813x_3^2 + 0{,}0021813x_3^2 \tag{6}$$

$$g_3(x) = 9{,}300961 + 0{,}0047026x_3x_5 + 0{,}0012547x_1x_3 + 0{,}0019085x_3x_4 \tag{7}$$

Con las siguientes restricciones:

$0 \leq g_1(x) \leq 92$, $90 \leq g_2(x) \leq 110$, $20 \leq g_3(x) \leq 25$

En el dominio limitado por:

$78 \leq x_1 \leq 100$, $33 \leq x_2 \leq 45$, $27 \leq x_{3,4,5} \leq 45$

3.2. Minimización del peso de un resorte de Tensión compresión

Este problema consiste en minimizar el peso de un resorte de tension/compresion que se encuentra sujeto a restricciones de defleccion mínima, esfuerzo cortante, frecuencia y diámetro externo, para ello se optimiza el diámetro del cable x_1, el diámetro medio del resorte x_2 y el numero de vueltas que tiene el resorte x_3 El problema a resolver es:

$$f(x) = (x_3 + 2)x_1^2 x_2 \tag{8}$$

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \tag{9}$$

$$g_2(x) = 1 - \frac{140{,}45 x_1}{x_2^2 x_3} \leq 0 \tag{10}$$

$$g_3(x) = \frac{x_1 + x_2}{1{,}5} - 1 \leq 0 \tag{11}$$

$$g_4(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \tag{12}$$

$$\tag{13}$$

En el dominio limitado por:

$0 \leq x_{1,2,3,4,5} \leq 15$ Cabe mencionar que este dominio se eligió de forma arbitraria ya que no se encontró una referencia con un dominio previamente delimitado.

3.3. Diseño de una olla a presión

El objetivo de esta función es el de diseñar un recipiente a presión que tenga el menor costo, este tiene 4 variables, El grueso de la lamina x_1, el grueso de la cabeza x_2, El radio interno x_3 y la longitud de la sección cilíndrica x_4, x_1 y x_2 son múltiplos enteros de 0.0625 mientras que las otras variables son continuas. La función a optimizar es:

$$f(x) = 0{,}6224 x_1 x_3 x_4 + 1{,}7781 x_2 x_3^2 + 3{,}1661 x_1 x_4^2 + 19{,}84 x_1 x_3^2 \tag{14}$$

$$g_1(x) = -x_1 + 0{,}0193 x_3 \leq 0 \tag{15}$$

$$g_2(x) = -x_2 + 0{,}00954 x_3 \leq 0 \tag{16}$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \tag{17}$$

En el dominio limitado por:

$1 * 0{,}0625 \leq x_1 \leq 100 * 0{,}0625$,$x_2 \leq 100 * 0{,}0625$, $0_3 \leq 100$, $0_4 \leq 250$. Al igual que en el caso anterior, el

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau'' x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1 x_2}, \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}$$

$$J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \delta(\vec{x}) = \frac{4PL^3}{Ex_3^3 x_4},$$

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

$$L = 14\ in \quad \delta_{\max} = 0.25\ in \quad E = 30 \times 10^6\ psi \quad G = 12 \times 10^6\ psi$$

dominio fue acotado de manera arbitraria.

3.4. Diseño de una viga soldada

La optimización de la viga consiste en minimizar el costo de producción de la viga considerando restricciones en distintas áreas, los valores a optimizar son: el ancho de la viga x_1 la longitud de la soldadura x_2 el ancho de la viga x_3 y finalmente el grosor de la viga x_4: La función a optimizar es:

$$f(x) = 1{,}1047x_1^2 x_2 + 0{,}04811x_3 x_4(14 + x_2) \tag{18}$$

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \tag{19}$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \tag{20}$$

$$g_3(x) = x_1 - x_4 \leq 0 \tag{21}$$

$$g_4(x) = 0{,}125 - x_1 \leq 0 \tag{22}$$

$$g_5(x) = \delta(x) - \delta_{max} \leq 0 \tag{23}$$

$$g_6(x) = P - P_c(x) \leq 0 \tag{24}$$

$$g_7(x) = 0{,}10471x_1^2 + 0{,}04811x_3 x_4(14{,}0 + x_2) - 5{,}0 \leq 0 \tag{25}$$

Las funciones necesarias para cumplir con estas condiciones, se presentan a continuación:

4. Resultados de las funciones

Una vez programadas tanto las funciones como el algoritmo de evolución diferencial se procedió a resolver las distintas funciones, en su mayoría se alcanzaron resultados similares a los obtenidos en las implementaciones del articulo sin embargo siguen sin tener la misma calidad, en la discusión se comentará mas al respecto. Los resultados que se obtuvieron son los siguientes:

Funcion	f($x_{aticulo}$)	f(x_{De})	Error porcentual
Himmelblau	-30988.951	-30752.9	-0.761726333
Resorte	0.012672	0.00479011	-62.19925821
Recipiente a presión	6059.79164	108673	1693.345489
Viga	1.726718	4.10482	137.7238206

Cuadro 1: Resultados de las ejecuciones

Se puede observar que solo en uno de los casos se logro llegar a un resultado mejor, sin embargo la diferencia es tan grande que se considera es un error dentro de la implementación, pese a que se revisó con suma atención, no se logró determinar cual era el error, por esta razón se reporta el resultado.

A pesar de ello, los valores en 3 de las 4 funciones son cercanos, lo que nos dice que si se realizaran algunas modificaciones al algoritmo sería posible refinar los resultados para alcanzar o incluso superar lo que se reporta en el artículo.

A continuación se presentan las gráficas de convergencia de las distintas ejecuciones.

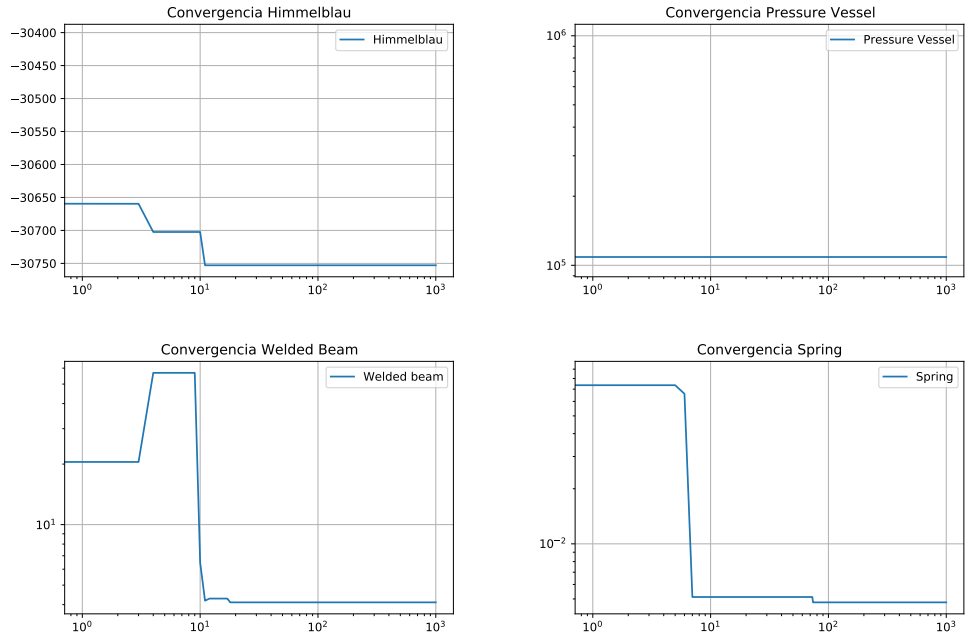

Figura 1: gráficas de las convergencias de las 4 funciones

5. Conclusión

como se puede observar en la tabla 1 y en las gráficas de la figura **??** El algoritmo es capaz de optimizar la mayoria de las funciones, sin embargo es claro que converge demasiado rápido a un mínimo local lo que genera

conflictos ya que pierde diversidad en pocas iteraciones, y eso limita la capacidad de exploración del dominio de búsqueda, lo que genera que las soluciones no sean de calidad, esto se podría resolver implementando algún método de control de diversidad, o reduciendo el criterio de elitismo dentro del algoritmo, probablemente si se realizan ambas cosas el resultado pueda tener una mejor calidad.

Referencias

[Chehouri et al., 2016] Chehouri, A., Younes, R., Perron, J., and Ilinca, A. (2016). A constraint-handling technique for genetic algorithms using a violation factor. *arXiv preprint arXiv:1610.00976*.