Estudio de la utilidad del conjunto de soluciones sub-óptimas en un Problema de Optimización Multiobjetivo.

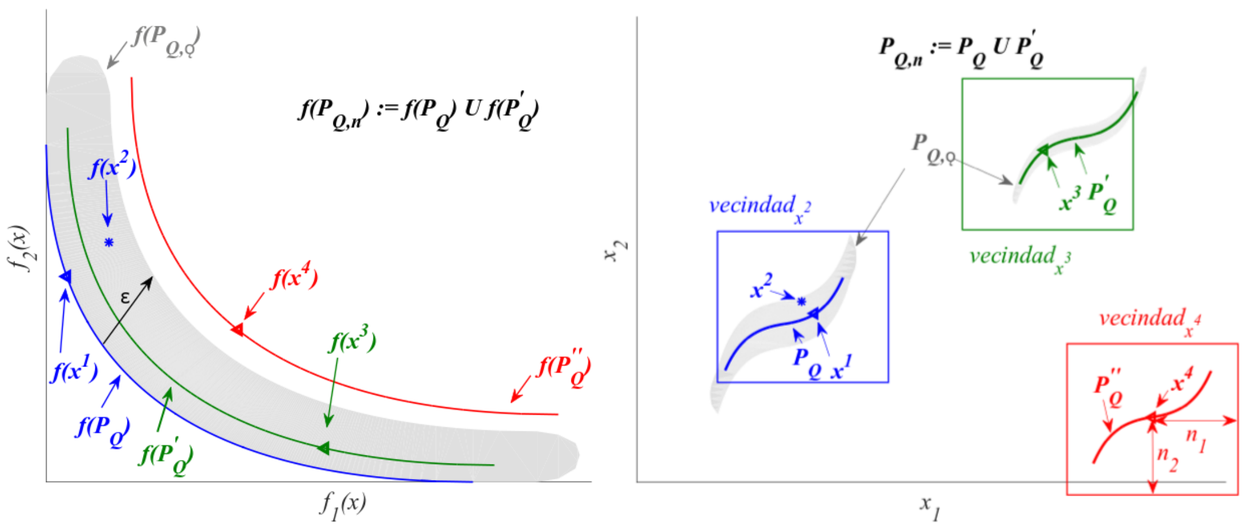
Alberto Pajares, Xavier Blasco, Javier Sanchis, Juan Manuel Herrero

{alpafer1, xblasco, jsanhis, juaherdu}@upv.es

IV SIMPOSIO CEA DE MODELADO, SIMULACIÓN y OPTIMIZACIÓN. VALLADOLID, 25-26 de Enero 2018.

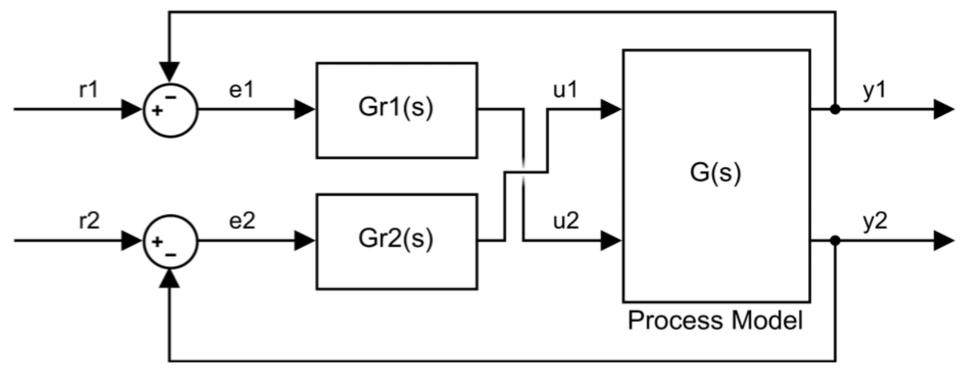
❑ Introducción:

- Un Problema de Optimización Multiobjetivo (MOP) busca caracterizar un frente de Pareto.
- Pero las soluciones multimodales o soluciones sub-óptimas significativamente diferentes (no vecinas) son soluciones de interés que podrían ser descartadas en un MOP clásico.
- Por ello se presenta en este trabajo un nuevo algoritmo (nevMOGA) que obtiene no solo las soluciones óptimas del frente de Pareto sino también las sub-óptimas no dominadas en su vecindad.
- Estas soluciones son potencialmente útiles para el diseñador por sus buenas prestaciones y diferenciación con el resto de soluciones. Su estudio puede informar de: la buena elección o no de los objetivos, sobre-parametrización del problema, etc.
- La utilidad de estas soluciones se observa claramente cuando se agregan objetivos de diseño para simplificar la etapa de decisión, ya que de esta forma aparece una multimodalidad artificial y se definen unas preferencias a priori al agregar objetivos.
- Para este nuevo algoritmo es necesario definir la vecindad (hasta cuando consideramos que dos soluciones son similares) en el espacio de parámetros y ϵ (que consideramos como soluciones sub-óptimas) en el espacio de objetivos.



Las soluciones óptimas y sub-óptimas no dominadas en su vecindad ($P_{Q,n}$) son los conjuntos P_Q y P_Q'. $\boldsymbol{x^2}$ es descartada porque esta dominada por una solución vecina $\boldsymbol{x^1}$. El conjunto P_Q'' es descartado porque no se consideran soluciones sub-óptimas.

❑ Ejemplo:



$$Y(s) = \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} = G(s) \begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix} \qquad G(s) = \begin{pmatrix} \frac{5e^{-40}}{100s+1} & \frac{1e^{-4}}{10s+1} \\ \frac{-5e^{-40}}{10s+1} & \frac{5e^{-40}}{100s+1} \end{pmatrix}$$

$$u_2(s) = Kc_1\left(e_1(s) + \frac{1}{Ti_1}\frac{1}{s}e_1(s)\right) \qquad u_1(s) = -Kc_2\left(e_2(s) + \frac{1}{Ti_2}\frac{1}{s}e_2(s)\right)$$

Definición del MOP:

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x})\ f_2(\boldsymbol{x})]$$

$$f_1 = \int_0^{t_f} e_1^2 + e_2^2\big|_{r_1=1, r_2=0}\, dt + \int_0^{t_f} e_1^2 + e_2^2\big|_{r_1=0, r_2=1}\, dt$$

$$f_2 = \int_0^{t_f} u_1^2 + u_2^2\big|_{r_1=1, r_2=0}\, dt + \int_0^{t_f} u_1^2 + u_2^2\big|_{r_1=0, r_2=1}\, dt$$

sujeto a: $f_1(\boldsymbol{x}) < 300, f_2(\boldsymbol{x}) < 200 \qquad \underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}} \qquad$ *Estable BC*

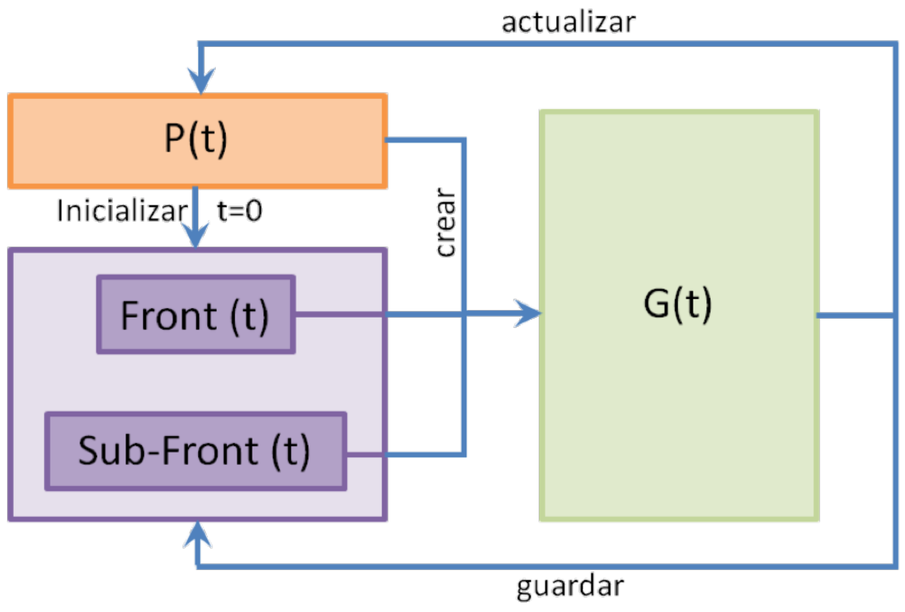
donde: $\boldsymbol{x} = [Kc_1, Ti_1, Kc_2, Ti_2] \qquad \mathbf{t_f} = 1000\ s$

$\overline{\boldsymbol{x}} = [-0.1, 200, 10, 100] \qquad \underline{\boldsymbol{x}} = [-1, 1, 0.1, 1]$

❑ Algoritmo[1]:

Algorithm 1. Pseudocódigo nevMOGA.

```
t:=0;
Front(t):= ∅;
Sub-Front(t):= ∅;
Crear Población Inicial P(t) aleatoriamente
Calcular f(∀ x ∈ P(t))
Ordenar Población P(t) usando índice niching
Inclusión de los individuos de P(t) en Front(t)
Inclusión de los individuos de P(t) no incluidos en Front(t) en Sub-Front(t)
for t:= 1:Número de iteraciones do
    Crea Población G(t)
    Calcular f(∀ x ∈ G(t))
    Inclusión de los individuos de G(t) en Front(t)
    Inclusión de los individuos de G(t) no incluidos en Front(t) en Sub-Front(t)
    Actualizar P(t) con los individuos de G(t)
    Ordenar Población P(t)
end for
```

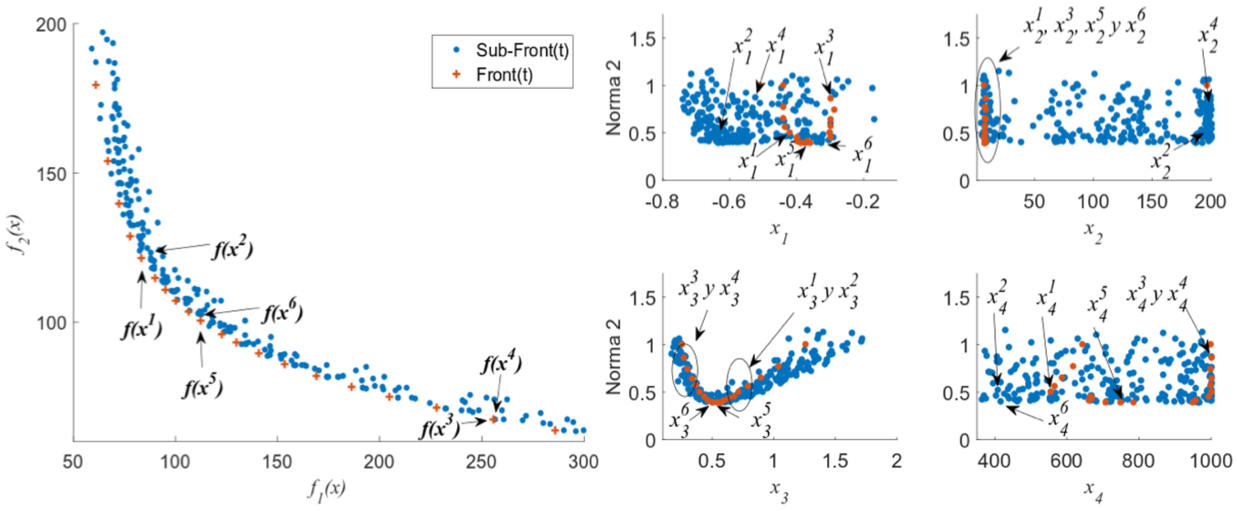


[1] Enviado a Complexity Journal (ISSN: 1099-0526)

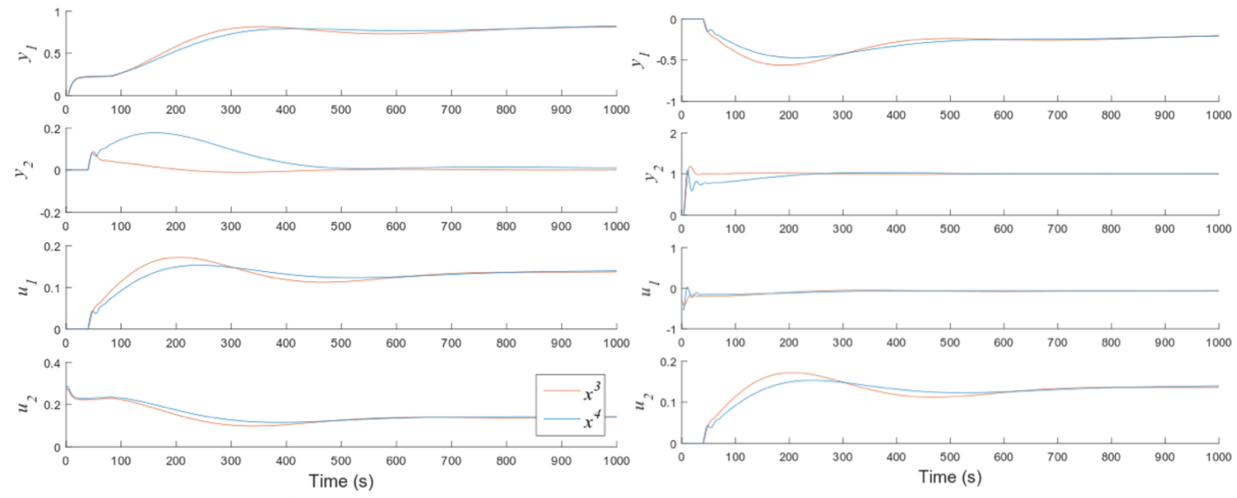
El algoritmo está compuesto de cuatro poblaciones:

- Población P(t) es la población principal y se encarga de explorar el espacio de búsqueda a lo largo de las iteraciones.
- Front(t) es el archivo donde se almacena el conjunto discreto de soluciones óptimas.
- Sub-Front(t) es el archivo donde se almacena el resto de soluciones del conjunto discreto P_{Qn}.
- G(t) es una población auxiliar que se utiliza para almacenar los nuevos individuos que se generan en cada iteración.

❑ Resultados:



Conjunto de soluciones óptimas y sub-óptimas (P_{Qn}) obtenido y representado en el espacio de parámetros mediante Level Diagrams. Las soluciones naranja son el conjunto óptimo.



Respuesta de las soluciones $\boldsymbol{x^3}$ y $\boldsymbol{x^4}$ .Estas soluciones tienen un rendimiento similar siendo soluciones no vecinas, por ello tienen un comportamiento diferente.

❑ Conclusiones:

- En este trabajo se presenta un nuevo algoritmo que tiene como objetivo encontrar las soluciones óptimas y también las soluciones sub-óptimas no dominadas en su vecindad.
- Estas soluciones nos permiten obtener mayor información para la etapa de decisión y con ello poder elegir la solución más idónea al problema con mayor criterio.
- En el ejemplo se observa como dos soluciones con rendimiento similar (aunque $\boldsymbol{x^1}$ domina a $\boldsymbol{x^2}$) obtienen una respuesta muy diferente (al ser soluciones no vecinas) lo que proporciona una mayor información al diseñador sobre nuevas alternativas de diseño que merecen estudio detallado.
- Por lo tanto, es útil encontrar el conjunto de soluciones sub-óptimas no dominadas en su vecindad y para ello se ha creado el algoritmo nevMOGA.

❑ Agradecimientos:

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (España) mediante el proyecto DPI2015-71443-R.