Nuevo enfoque algorítmico para la mejora en convergencia y dispersión de algoritmos evolutivos biobjetivos que aproximan al completo el frente de Pareto.

Antonio Borrego Ortega
Mariano Luque Gallego
Rubén Saborido Infantes

I Iberian Conference on MCDM/A (IMCDM/A)

mayo de 2025

Indice

Indice

¿Qué es un problema de optimización multiobjetivo (MOP)?

Se define como

$$\begin{array}{ll} \text{maximizar} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \\ \text{s.a} & \mathbf{x} \in \Omega \subseteq \mathbb{R}^m, \end{array} \tag{1}$$

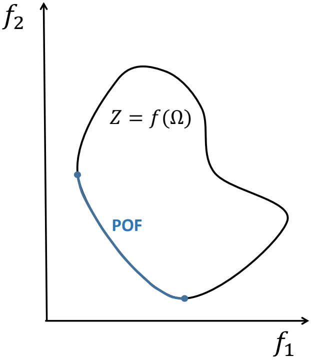
- $\mathbf{x}$ es una solución m-dimensional en la región factible Ω.
- $Z = f(\Omega)$ constituye el espacio objetivo.
- $\mathbf{f} : \Omega \to \mathbb{R}^k$ es un vector $k-$ dimensional en el espacio objetivo $\mathbb{R}^k$.

Conceptos Básicos

Solución Eficiente y frente de Pareto óptimo (POF)

Una solución $\mathbf{x} \in \Omega$ se dice que domina a una solución $\bar{\mathbf{x}} \in \Omega$ si:

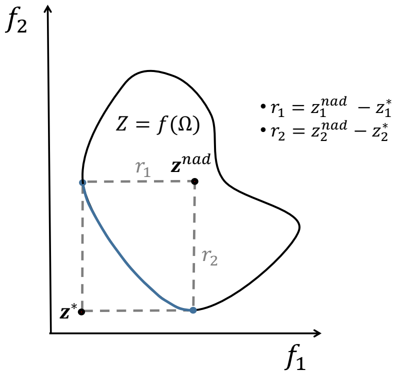
- $f_i(\mathbf{x}) \leq f_i(\bar{\mathbf{x}})$ para todo $i = 1, \ldots, k$, $f_j(\mathbf{x}) < f_j(\bar{\mathbf{x}})$
- $\mathbf{x}$ es una solución **eficiente** si no existe ninguna solución $\bar{\mathbf{x}} \in \Omega$ que la domine.



- El conjunto de todas las soluciones eficientes se denomina conjunto eficiente (**E**).
- La imagen de dicho conjunto por f , $f(E)$, se denomina **POF**

Vector ideal y vector nadir

- *Vector ideal* $\mathbf{z}^* = (z_1^*, z_2^*, \ldots, z_k^*)^T$ es el vector objetivo con $z_i^* = \min_{\mathbf{x}\in E} f_i(\mathbf{x}) = \min_{\mathbf{x}\in \Omega} f_i(\mathbf{x})$, $i = 1, 2, \ldots, k$.
- *Vector nadir* $\mathbf{z}^{nad} = (z_1^{nad}, z_2^{nad}, \ldots, z_k^{nad})^T$ es el vector objetivo con $z_i^{nad} = \max_{\mathbf{x}\in E} f_i(\mathbf{x})$, $i = 1, 2, \ldots, k$.



Función de logro de Wierbizcki

- Punto de referencia: $\mathbf{q} = (q_1, \ldots, q_k)^T$.
- Vector de pesos: $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^T$, $\mu_i > 0$ $(i = 1, \ldots, k)$
- $\rho > 0$, que debe ser un valor pequeño, es el denominado término aumentado.

$$s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \boldsymbol{\mu}) = \max_{i=1,\ldots,k} \{\, \mu_i(f_i(\mathbf{x}) - q_i) \,\} + \rho \sum_{i=1}^{k} \mu_i(f_i(\mathbf{x}) - q_i) \quad (2)$$

Aplicación

Fijados un punto $\mathbf{q}$ y un vector de pesos $\boldsymbol{\mu}$, obtener $\min_{\mathbf{x}\in\Omega} s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \boldsymbol{\mu})$ equivale a encontrar la una solución eficiente asociada a dicho punto y dicho vector.

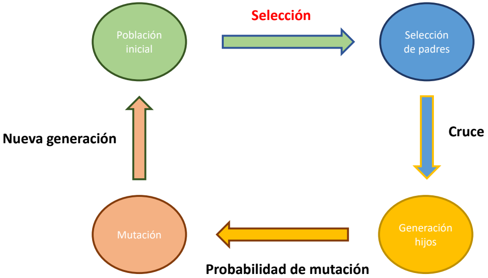
¿Qué es un Algoritmo Evolutivo?

- Un **algoritmo evolutivo** es un algoritmo inspirado en el proceso de la evolución biológica.
- La rama que estudia la aplicación de dichos algoritmos en optimización multiobjetivo se conoce como **EMO**.
- Los algoritmos EMO son enfoques basados en población.
- Estos algoritmos buscan encontrar soluciones óptimas a problemas complejos mediante procesos de **selección**, **mutación** y **reproducción**.
- El objetivo es mejorar gradualmente las soluciones a lo largo de un conjunto de generaciones.

Esquema flujo EMO

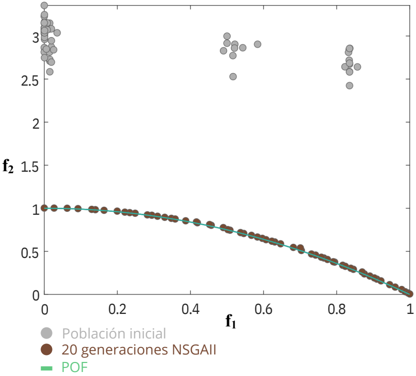
Pasos en un EMO

1. Inicialización
2. Evaluación
3. Selección
4. Recombinación y Mutación
5. Reemplazo
6. Repetir



Objetivos de un EMO sin preferencias

- Convergencia sobre el frente de Pareto del problema.
- Diversidad en los óptimo de Pareto que se obtienen para mostrar una distribución uniforme del POF.



Marco Algorítmico AS-MOEA

- Marco algorítmico que permita establecer un **equilibrio** entre convergencia y diversidad.
- **Adaptable** a cualquier EMO sin preferencias.
- Mejore o mantenga el comportamiento del EMO de partida en términos de convergencia, dispersión y uniformidad de soluciones.
- Problemas bi-objetivos.

AS-MOEA

Estructura general

- **Fase 1**: Estudiar la aproximación del POF obtenida inicialmente por el EMO y dividirla en subregiones.
- **Fase 2**: Continuar el proceso de optimización del tramo del POF que encontramos en cada subregion de manera local aplicando al función de logro.

Notación

- N, tamaño de la población.
- P^h, población en el espacio objetivo obtenida en la generación h del algoritmo.
- $NG_T = NG_{T_1} + NG_{T_2}$, $NG_{T_1} = p \cdot NG_T$, $0 < p < 1$.

Enfoque algorítmico: Fase 1

Objetivo Fase 1

- Obtener una aproximación inicial del POF.
- Identificar subregiones dentro de la aproximación.

Generaciones Fase 1

$NG_{T_1} = p \cdot NG_T$ generaciones del algoritmo

No demasiado pequeño $\rightarrow$ obtener una buena aproximación inicial

No demasiado grande $\rightarrow$ que resten generaciones para la **Fase 2**.

Enfoque algorítmico: Fase 1

División en subregiones

1. Se divide r_1 en N_s intervalos de misma longitud.
2. Estimamos el punto ideal y nadir de cada intervalo, $\mathbf{z^{l,r}} = (z_1^{l,r}, z_2^{l,r})$ y $\mathbf{z^{u,r}} = (z_1^{u,r}, z_2^{u,r}), r = \{1, \ldots, N_s\}$.

Finalmente se define para $r = \{1, \ldots, N_s\}$:

$$S_r = \{\mathbf{z} = (z_1, z_2) \in Z, \text{ tal que } z_i^{l,r} \leq z_i \leq z_i^{u,r}, \ i = 1, 2\},$$

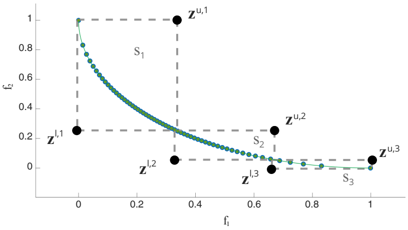


Figura: Aproximación de POF del problema IDTLZ2 por el algoritmo NSGA-III con $N = 60$ tras $NG_{T_1} = 250$ generaciones.

Enfoque algorítmico: Fase 2

Objetivo Fase 2

- Mejorar en términos de dispersión, convergencia y uniformidad la aproximación.
- Aproximar la parte de POF que pertenece a cada subregion localmente.

Parámetros para cada S_r

- Punto de referencia, $\mathbf{q^r}$.
- Muestra de vectores de peso, $\boldsymbol{\mu}^{r,j}$, $j = 1, \ldots, n_r$.
- $\displaystyle\sum_{r=1}^{N_s} n_r = N$.

Enfoque algorítmico: Fase 2

Cálculo de $\boldsymbol{\mu}^{r,j}$, $j = 1, \ldots, n_r$

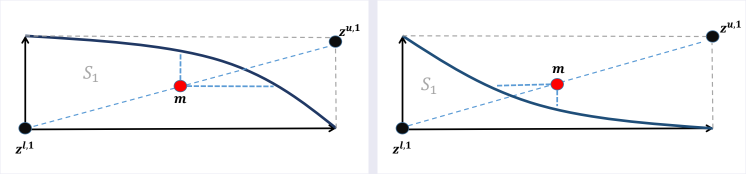
1. Se decide qué cantidad de vectores de peso (n_r) se utilizan para aproximar cada S_r.
2. Se genera n_r de vectores igualmente distribuidos en $(0, 1)^2$.
3. Se ajustan $\mu_1^{r,j}$ y $\mu_2^{r,j}$ con los rangos de variación de los objetivos en dichas subregiones.

Enfoque algorítmico: Fase 2

Cálculo de $\boldsymbol{\mu}^{r,j}$, $j = 1, \ldots, n_r$

1. Se decide qué cantidad de vectores de peso (n_r) se utilizan para aproximar cada S_r.
2. Se genera n_r de vectores igualmente distribuidos en $(0,1)^2$.
3. Se ajustan $\mu_1^{r,j}$ y $\mu_2^{r,j}$ con los rangos de variación de los objetivos en dichas subregiones.

Cálculo de q^r



(a) $\mathbf{q}^1 = \mathbf{z}^{l,1}$.

(b) $\mathbf{q}^1 = \mathbf{z}^{u,1}$.

Enfoque algorítmico: Fase 2

Clasificación en fronteras

1. Aplicamos operadores de mutación y cruce sobre P^h generando N individuos nuevos y consideramos la población combinada P de $2N$ individuos.
2. Calculamos para todo $\mathbf{x} \in P$ el valor de $s(\mathbf{q}^r, f(\mathbf{x}), \boldsymbol{\mu}^{r,j})$.
3. Incluimos en F_1^h la solución que minimiza dicho valor y la eliminamos temporalmente de P
4. Repetimos dicho proceso para cada vector asociado a cada subregion.
5. $F_1^h = P^{h+1}$

Ajuste de subregiones

Abarcar zonas del POF no obtenidas en la aproximación inicial del EMO.

Enfoque algorítmico: Fase 2

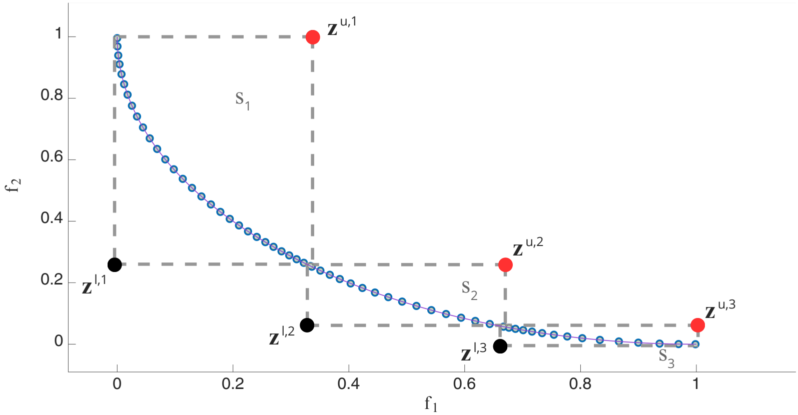
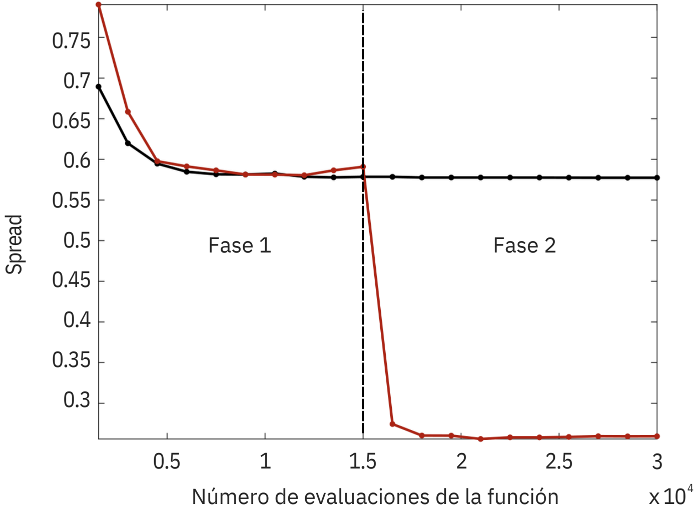


Figura: Aproximación de POF del problema IDTLZ2 aplicando AS-MOEA NSGAIII con $N = 60$ tras 375 generaciones (250 aplicando Fase 1 y 125 aplicando Fase 2).

Enfoque algorítmico: Fase 2



NSGAIII

ASMOEA-NSGAIII

Experimentos

Métricas

- Hypervolume (HV).
- *Spread*.
- Spacing (Sp).

Parámetros

- Tamaño de la población: $N = 180$.
- Número máximo de generaciones: $h_{max} = 200$.
- Número de veces que ajustamos las subregiones: $n_a = 10$.
- Número de subregiones consideradas: $N_s = 3$.
- Porcentaje de generaciones aplicadas a la cada fase: 50 % ($p = 0.5$).

Problemas

Un total de 22 problemas test que proporciona PlatEMO. (ZDT, IDTLZ, DTLZ, WFG).

Resultados

- Comparativa NSGAIII con ASMOEA-NSGAIII:

Métrica	+	=	-
HV	10	7	5
Spread	11	3	8
Sp	14	1	7

- Comparativa RVEA con ASMOEA-RVEA:

Métrica	+	=	-
HV	21	0	1
Spread	15	4	3
Sp	20	1	1

Trabajo futuro

Próximos enfoques

- Estudiar como definir las subregiones en problemas de 3, 5 y 8 objetivos respectivamente.
- Estudiar detalladamente el cómo escoger el número de generaciones asignados a cada Fase.
- Realizar pruebas para más algoritmos que aproximen el frente al completo.
- Estudiar como afecta el incremento en el número de subregiones para los diferentes problemas.
- Ajustar las subregiones para ampliar los extremos en caso de no tener una buena aproximación.