

Optimización Multimodal y Multiobjetivo

Sergio Acosta
Universidad Nacional Bogotá, Colombia
Email:sfacostale@unal.edu.co

Resumen—Los métodos de optimización multimodal y multiobjetivo intentan descubrir diferentes regiones de interés. En particular los métodos basados en algoritmos genéticos buscan mantener diversidad en la población con el ánimo de promover la detección de diferentes mínimos o máximos locales así como la detección de fronteras de Pareto. En ese documento se relacionan los resultados de la implementación de 3 de estos métodos.

máximo en color azul, la desviación estándar del *fitness* en color rojo, la mediana en color verde, y la media en color negro.

II-A. Función $M1$

$$M_1(x) = \sin^6(5x\pi)$$

I. INTRODUCCIÓN

Los métodos de optimización multimodal y multiobjetivo intentan descubrir diferentes regiones de interés. En particular los métodos basados en algoritmos genéticos buscan mantener diversidad en la población con el ánimo de promover la detección de diferentes mínimos o máximos locales así como la detección de fronteras de Pareto. En ese documento se relacionan los resultados de la implementación en la sección II del método de optimización multimodal *Self Adaptation of Operator Rates for Multimodal Optimization* [4]. Este es aplicado a 4 funciones con máximos locales de diferentes características. En la sección III se presentan los resultados de la implementación de los algoritmos NSGA-II [6] y SPEA-2 [5]. En la sección IV se relacionan algunas conclusiones de respectó a las implementaciones. El código asociado a este documento así como a diferentes temas de computación evolutiva se encuentra disponible en el repositorio relacionado en la sección ANEXO.

II. MULTIMODAL

Las funciones a optimizar son la siguientes.

$$M_1(x) = \sin^6(5x\pi)$$

$$M_2(x) = \exp^{-2\ln(2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5x\pi)$$

$$M_3(x) = \sin^6(5\pi[x^0,75 - 0,05])$$

$$M_4(x) = \exp^{-2\ln(2)\left(\frac{x-0.08}{0.854}\right)^2} \sin^6(5\pi[x^0,75 - 0,05])$$

Figura 1: Funciones de experimentación

El método usado esta basada en el algoritmo HAEA [2] en combinación con el método *Deterministic Crowding*. Cada ejecución consistió de 250 iteraciones y 64 individuos. Además de las gráficas que presentan la distribución de los individuos cada 50 iteraciones, estas gráficas son complementadas con una gráfica que presenta el comportamiento del operador genético usado. Azul flip-mutación, rojo cruce y verde mutación. Una codificación de 16 bits fue usada. La ultima gráfica de cada conjunto presenta la evolución del *fitness* mínimo y

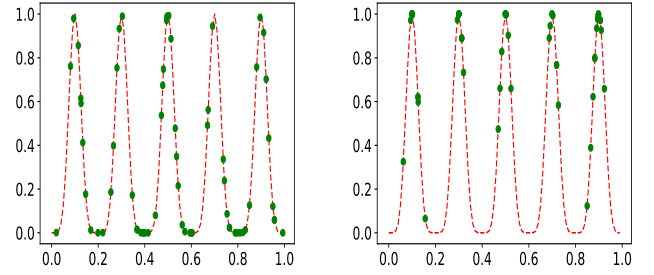


Figura 2: Left: Initial population distribution. Right: Population distribution after 50 iterations

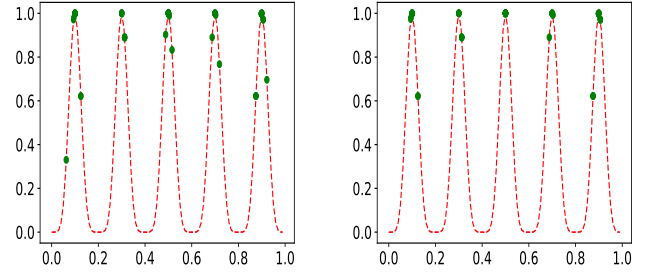


Figura 3: Left: Population distribution after 100 iterations. Right: Population distribution after 150 iterations

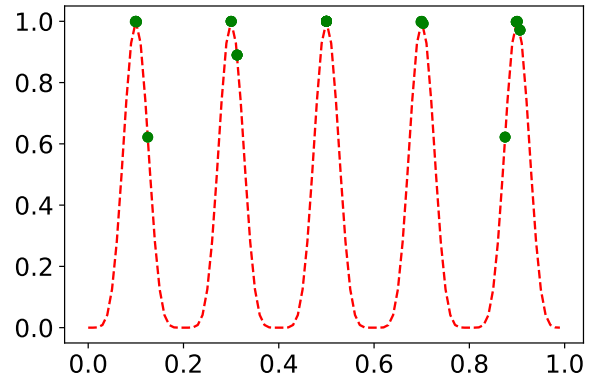


Figura 4: Population distribution after 200 iterations

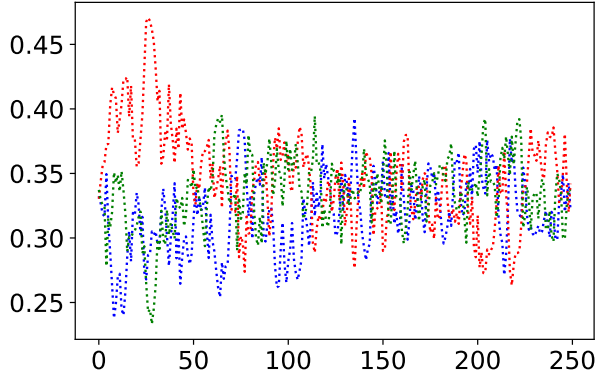


Figura 5: Selección de operadores genéticos

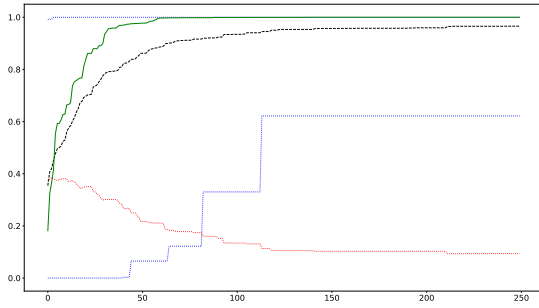


Figura 6: Convergencia

II-B. Función M2

$$M_2(x) = \exp^{-2 \ln(2) \left(\frac{x-0.1}{0.8} \right)^2} \sin^6(5x\pi)$$

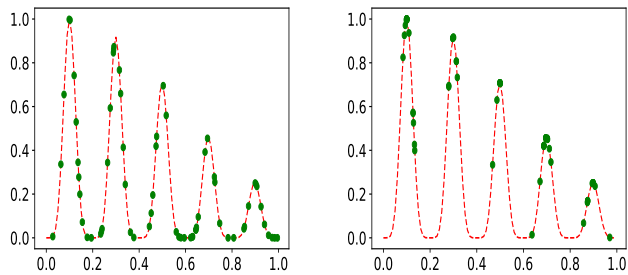


Figura 7: Left: Initial population distribution. Right: Population distribution after 50 iterations

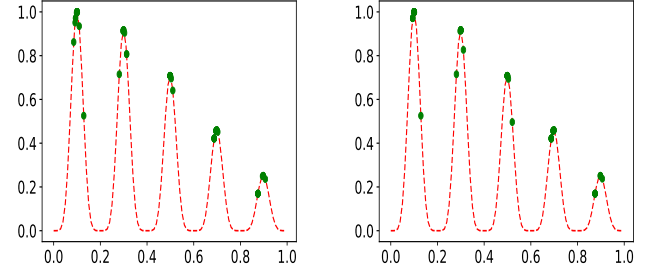


Figura 8: Left: Population distribution after 100 iterations. Right: Population distribution after 150 iterations

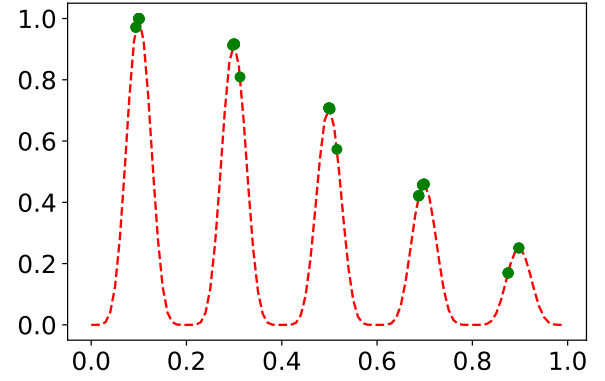


Figura 9: Population distribution after 200 iterations

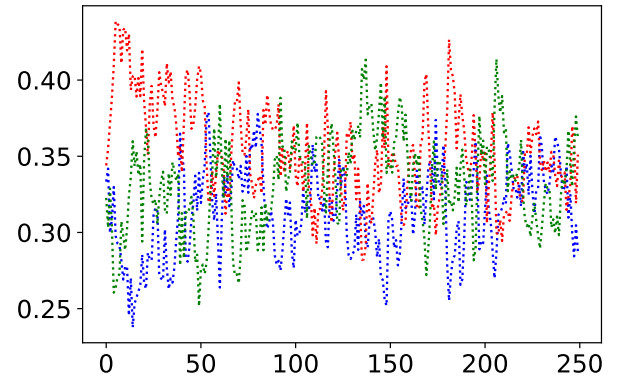


Figura 10: Selección de operadores genéticos

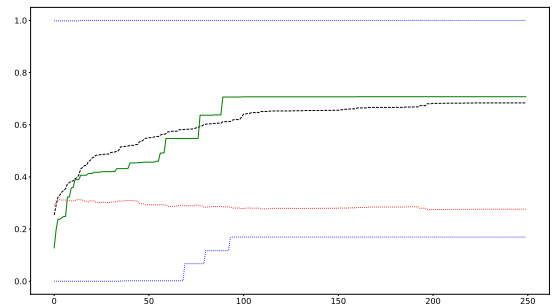


Figura 11: Convergencia

II-C. Función M3

$$M_3(x) = \sin^6(5\pi[x^0,75 - 0,05])$$

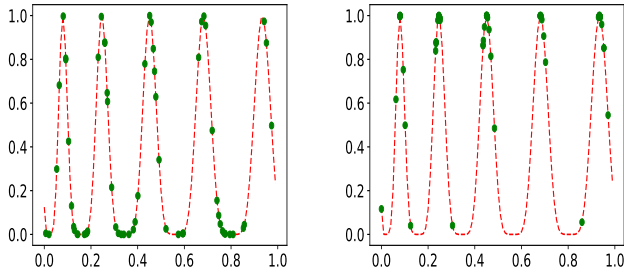


Figura 12: Left: Initial population distribution. Right: Population distribution after 50 iterations

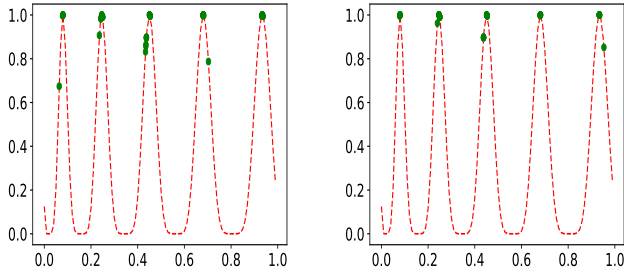


Figura 13: Left: Population distribution after 100 iterations. Right: Population distribution after 150 iterations

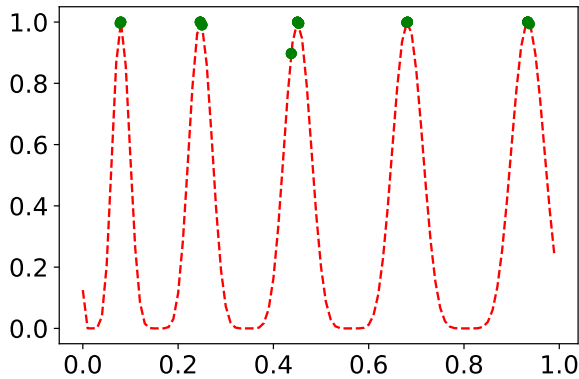


Figura 14: Population distribution after 200 iterations

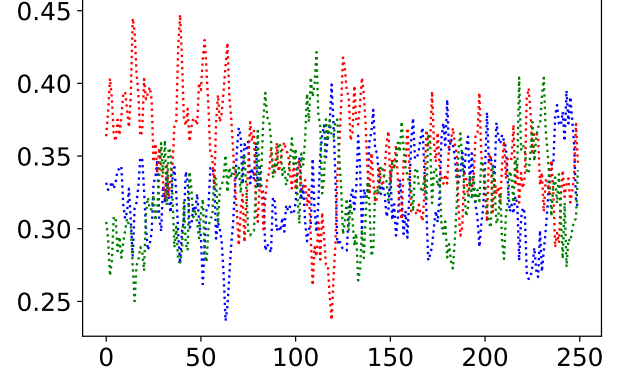


Figura 15: Selección de operadores genéticos

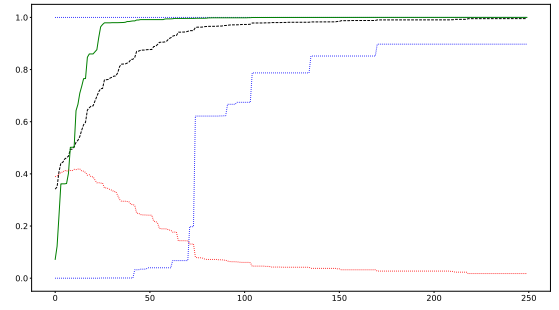


Figura 16: Convergencia

II-D. Función M4

$$M_4(x) = \exp^{-2 \ln(2) \left(\frac{x-0,08}{0,854} \right)^2} \sin^6(5\pi[x^0,75 - 0,05])$$

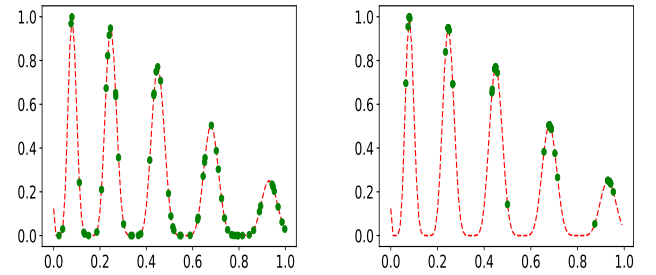


Figura 17: Left: Initial population distribution. Right: Population distribution after 50 iterations

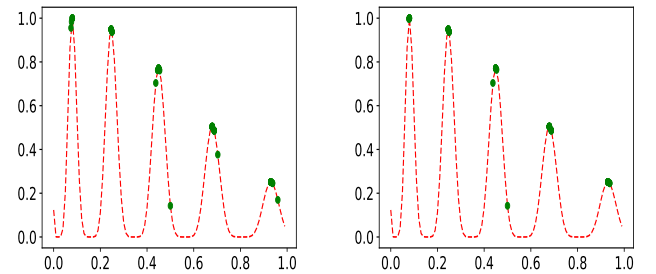


Figura 18: Left: Population distribution after 100 iterations. Right: Population distribution after 150 iterations

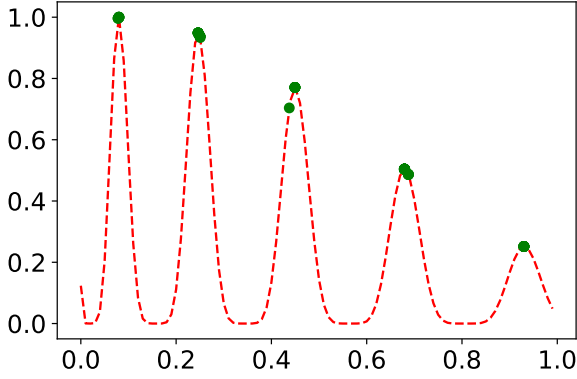


Figura 19: Population distribution after 200 iterations

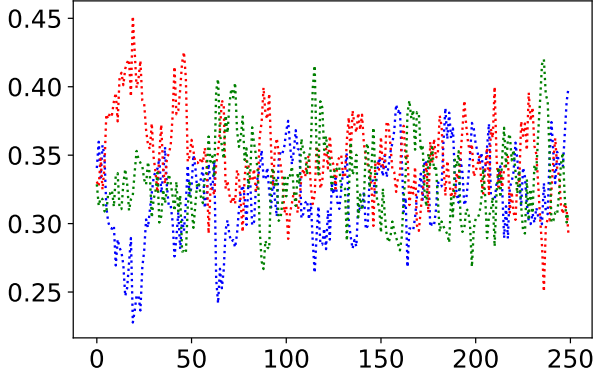


Figura 20: Selección de operadores genéticos

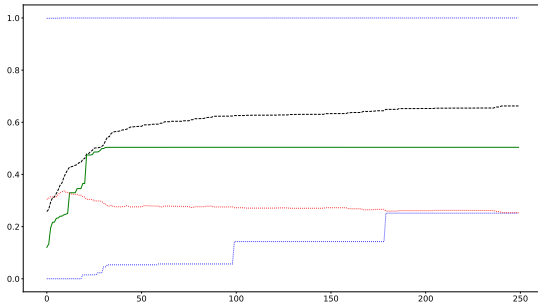


Figura 21: Convergencia

III. MULTIOBJETIVO

Tanto NSGA-II [6] y SPEA-2 [5] han sido evaluados con tres pares de funciones continuas.

Figura 22 Funciones de experimentación multiobjetivo

n	Funciones Objetivo	Dominio
100	$SPH - 2$ $f_j(x) = \sum_{1 \leq i \leq n, i \neq j} x_i^2 + (x_j - 1)^2$ $1 \leq j \leq m, m = 2$	$[-10^3, 10^3]$
3	KUR $f_1(x) = \sum_{i=1}^{n-1} -10 \exp^{-0.2 \sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x) = \sum_{i=1}^n (x_i ^{0.8} + 5 \sin^3 x_i)$	$[-5, 5]$
10	$ZDT6$ $f_1(x) = 1 - \exp^{-4x_1} \sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (\frac{f(x)}{g(x)})^2]$ $g(x) = 1 + 9[\frac{(\sum_{i=2}^n x_i)}{(n-1)}]^{0.25}$	$[0, 1]$

III-A. NSGA2

La población en cada experimento fue de 50 individuos con una codificación de 16 bits. Se usó el algoritmo genético canónico con una probabilidad de cruce del 70 % y una probabilidad inversamente proporcional a la longitud del código genético.

III-A1. SPH2 con NSGA2:

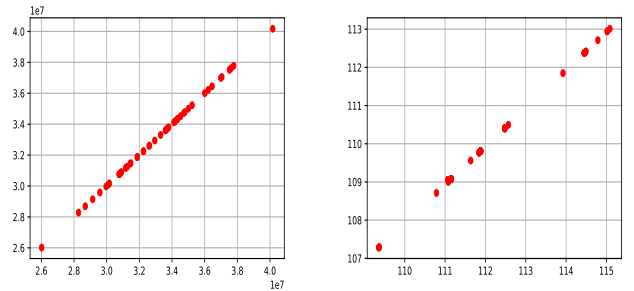


Figura 23: Left: Initial population distribution. Right: Front distribution after 1200 iterations

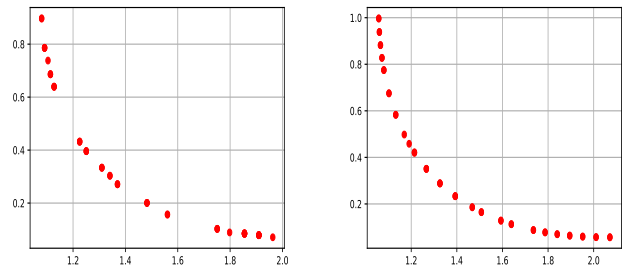


Figura 24: Left: Front distribution after 2400 iterations. Right: Front distribution after 3600 iterations

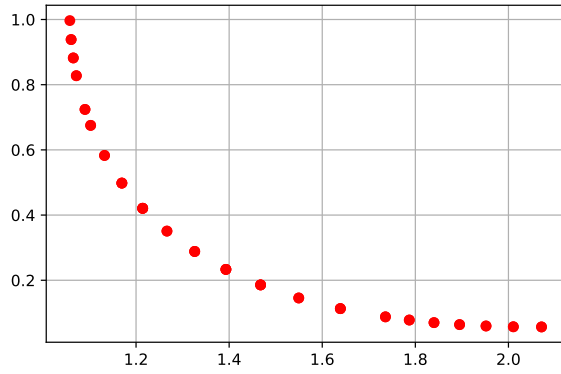


Figura 25: Final front distribution after 4800 iterations

III-A2. KUR con NSGA2:

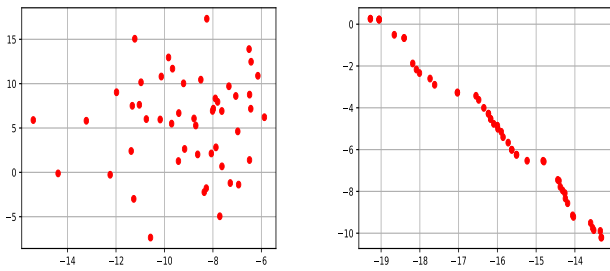


Figura 26: Left: Initial population distribution. Right: Front distribution after 30 iterations

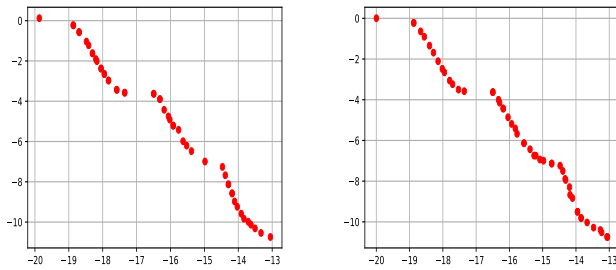


Figura 27: Left: Front distribution after 60 iterations. Right: Front distribution after 90 iterations

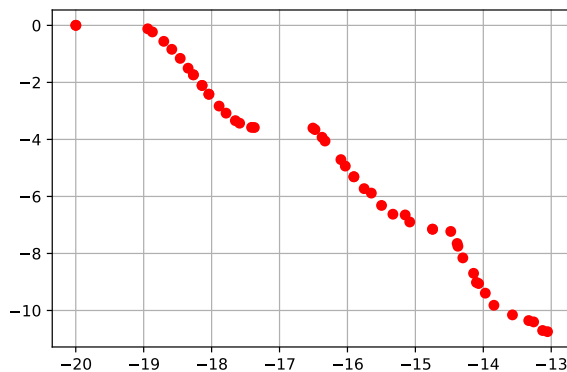


Figura 28: Final front distribution after 120 iterations

III-A3. ZDT6 con NSGA2:

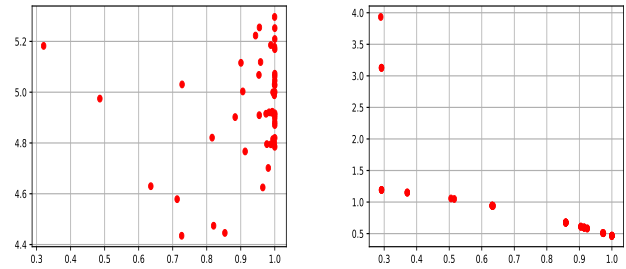


Figura 29: Left: Initial population distribution. Right: Front distribution after 25 iterations

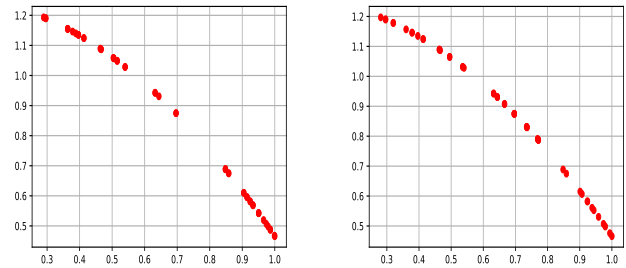


Figura 30: Left: Front distribution after 50 iterations. Right: Front distribution after 75 iterations

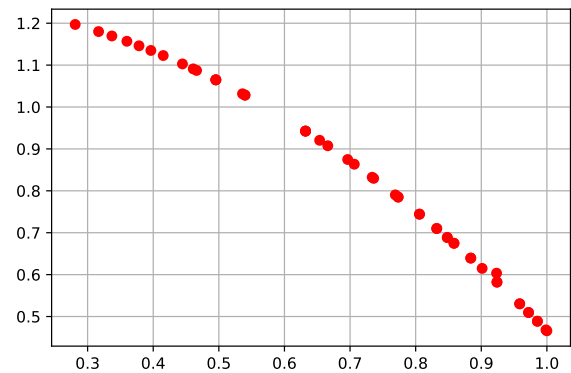


Figura 31: Final front distribution after 100 iterations

III-B. SPEA2

La población en cada experimento fue de 50 individuos con una codificación de 16 bits. Se usó el algoritmo genético canónico con una probabilidad de cruce del 70 % y una probabilidad inversamente proporcional a la longitud del código genético.

III-B1. SPH2 con SPEA2:

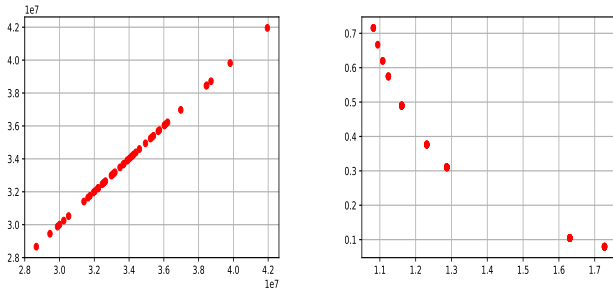


Figure 32: Left: Initial population distribution. Right: Front distribution after 3000 iterations

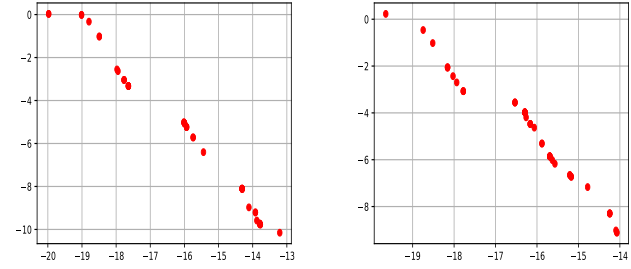


Figure 36: Left: Front distribution after 800 iterations. Right: Front distribution after 1200 iterations

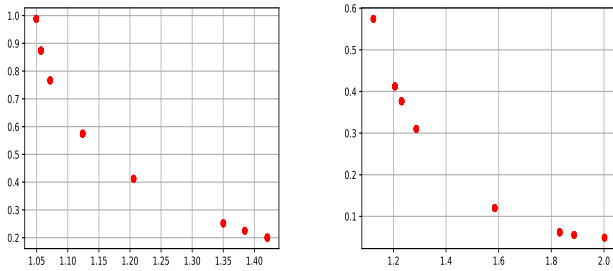


Figure 33: Left: Front distribution after 6000 iterations. Right: Front distribution after 9000 iterations

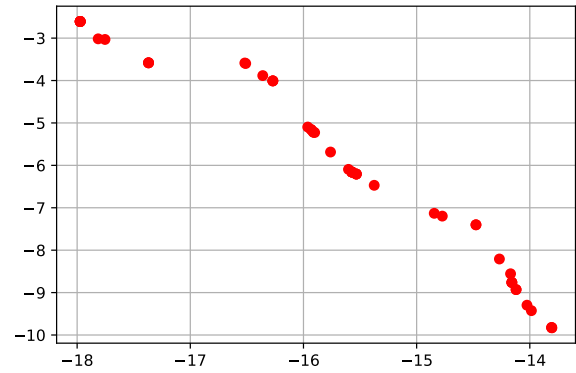


Figure 37: Final front distribution after 1600 iterations

III-B3. ZDT6 con SPEA2:

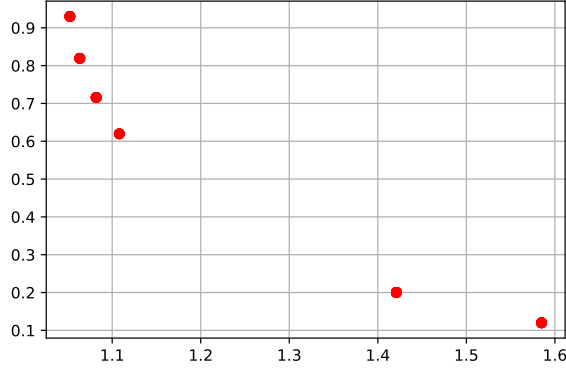


Figure 34: Final front distribution after 12000 iterations

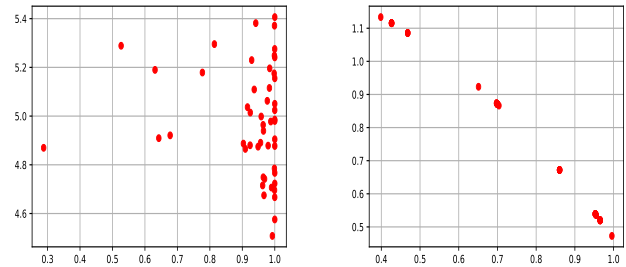


Figure 38: Left: Initial population distribution. Right: Front distribution after 400 iterations

III-B2. KUR con SPEA2:

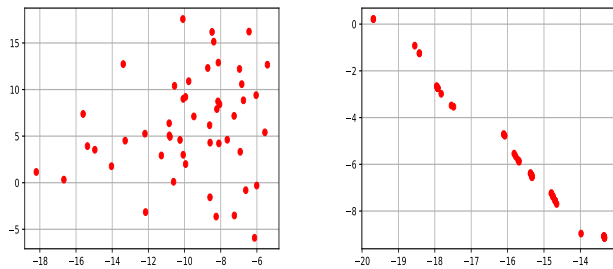


Figure 35: Left: Initial population distribution. Right: Front distribution after 400 iterations

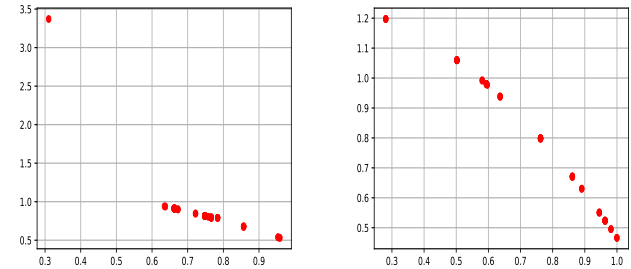


Figure 39: Left: Front distribution after 800 iterations. Right: Front distribution after 1200 iterations

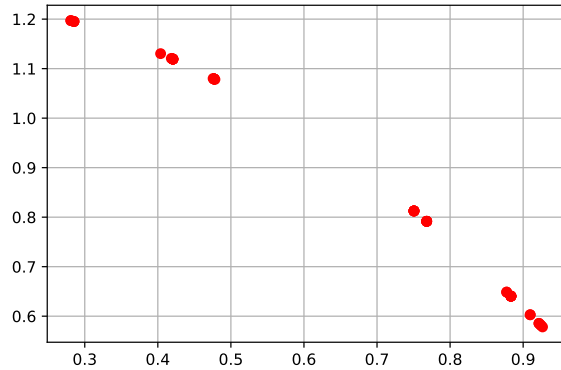


Figura 40: Final front distribution after 1600 iterations

IV. CONCLUSIONES

La implementación de los algoritmos presentados en este documento se realizó en Python y el paquete de cómputo científico NumPy. La implementación del algoritmo presentado en [4], se considera exitosa dado que para cada una de las funciones M1, M2, M4 y M4 los diferentes máximos locales son detectados. Respecto a los algoritmos de optimización multiobjetivo la implementación del algoritmo NSGA-II resultó más satisfactoria, ya que esta convergió para los tres experimentos en poblaciones mejor distribuidas sobre la frontera de Pareto detectada. Para la implementación del algoritmo SPEA2, en los tres experimentos la población resultante no tiene la suficiente diversidad para lograr una distribución amplia sobre la frontera detectada.

V. ANEXO

<https://github.com/sergiojx/minmethods>

REFERENCIAS

- [1] Haupt, Randy L. and Haupt, Sue Ellen. *Practical Genetic Algorithms Chapter 2 The Binary Genetic Algorithm*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [2] Gomez J. (2004) *Self Adaptation of Operator Rates in Evolutionary Algorithms*. In: Deb K. (eds) Genetic and Evolutionary Computation – GECCO 2004. GECCO 2004. Lecture Notes in Computer Science, vol 3102. Springer, Berlin, Heidelberg
- [3] Beyer, HG. & Schwefel, HP. *Evolution strategies – A comprehensive introduction*. Natural Computing (2002) 1: 3. <https://doi.org/10.1023/A:1015059928466>
- [4] Gomez J. (2004) *Self adaptation of operator rates for multimodal optimization*. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)
- [5] Eckart Zitzler and Marco Laumanns and Lothar Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, 2001
- [6] Kalyanmoy Deb and Samir Agrawal and Amrit Pratap and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, 2002