Revista Publicando, 3(8). 2016, 37-51. ISSN 1390-9304

Optimización continua con dos estrategias evolutivas: Coma (μ, λ) y Plus (μ + λ)

Cristian Zambrano Vega[1], Joel A. Cedeño Muñoz[2], Jefferson Bravo Salvatierra[3], Eduardo Díaz Ocampo[4]

1 Universidad Técnica Estatal de Quevedo, *czambrano@uteq.edu.ec*

2 Universidad Técnica Estatal de Quevedo, *jacedeno@uteq.edu.ec*

3 Universidad Técnica Estatal de Quevedo, *jbravo@uteq.edu.ec*

4 Universidad Técnica Estatal de Quevedo, *ediaz@uteq.edu.ec*

RESUMEN

El objetivo del presente trabajo es implementar un estudio experimental comparativo de dos estrategias evolutivas de reemplazo: Estrategia Coma (μ, λ) y Estrategia Plus (μ + λ) con Mutación Auto-Adaptativa, pasos independientes y no correlacionados para cada variable, en un algoritmo evolutivo que resuelve el problema de la esfera, logrando minimizar la función hasta un valor óptimo de $f(\vec{x}) \leq 10^{-6}$, con el objetivo de conocer el esfuerzo computacional necesario (medido en número de evaluaciones) para lograr el valor óptimo establecido, y definir diferencias significativas entre los resultados obtenidos de ambas estrategias. Se empelaron dos tipos de recombinación para la generación de nuevas soluciones, Discreta y Aritmética. Los resultados indicaron que mayormente no existen diferencias significas entre ambas estrategias sobre los diferentes tamaños del problema, por lo que basados en el esfuerzo computacional, la estrategia Plus converge más rápidamente al valor óptimo que la estrategia Coma, concluyendo que esta estrategia brinda mejores resultado inclusos sobre dimensiones del problemas con mayor valor.

PALABRAS CLAVE: Optimización, Optimización Continúa, Estrategias evolutivas, Minimización, Algoritmos evolutivos.

Revista Publicando, 3(8). 2016, 37-51. ISSN 1390-9304

Continuous Optimization with two Evolutionary Strategies: Comma (μ, λ) and Plus (μ + λ)

ABSTRACT

In the present work we have carried out an experimental study of two Evolutionary Strategies: Comma (μ, λ) and Plus (μ + λ) with Auto-Adaptive mutation, independent and uncorrelated steps for each variable, in an evolutionary algorithm that solves the optimization problem of the Sphere, minimizing the objective function to an optimal value of $f(\vec{x}) \leq 10^{-6}$; with the aim of know the required computational effort (measured in number of evaluations) to achieve the optimal value, and define significant differences between the results of both strategies. To the generation of new solutions, two types of recombinations were applied: Discreet and Arithmetic. The results indicated that mostly there is not significant differences between the two strategies on the different size of the problem, so based on the computational effort, the Plus strategy converges faster to the optimal value than Comma strategy, concluding that this strategy provides better results even on high dimensions of the problem.

KEYWORDS: Optimization, Continuous Optimization, Evolutionary Strategies, Evolutionary algorithms.

Revista Publicando, 3(8). 2016, 37-51. ISSN 1390-9304

1. INTRODUCCIÓN

1.1. ESTRATEGIAS EVOLUTIVAS

Las estrategias evolutivas (Beyer, H. G., & Schwefel, H. P., 2002) fueron desarrolladas a principios de los 70 por Rechenberg y Schwefel como un método de resolución de problemas de optimización de ingeniería. Entre sus características destacables:

- ✓ Rápidas.
- ✓ Muy adecuadas para la optimización continua.
- ✓ Bien conocidas desde el punto de vista teórico.
- ✓ Incluyen como mecanismo estándar la auto-adaptación de los parámetros de la mutación (Salazar-Horning, E.J., Rojas-Oyarzún, R.S., 2010).

Las estrategias evolutivas se enmarcan dentro del paradigma de los algoritmos evolutivos. La principal diferencia de las estrategias evolutivas con respecto a los otros algoritmos evolutivos es que en las estrategias evolutivas únicamente se usa el operador de mutación, es decir, no hay intercambio de información entre padres para tener un descendiente, lo cual efectivamente ocurre en otros paradigmas como los algoritmos genéticos o la programación genética. En las estrategias evolutivas el proceso de engendrar produce los **λ** hijos a partir de los **μ** padres mediante mutación.

Dentro de las estrategias evolutivas se defines los siguientes parámetros:

- **μ** Tamaño de la población inicial
- **λ** Tamaño de la población descendente
- **ρ** Tamaño de la familia (padres) ($1 \leq \boldsymbol{\rho} \leq \mu$)

A continuación detallaremos las dos estrategias en estudio

1.2. LA ESTRATEGIA EVOLUTIVA: PLUS (μ + λ)

En esta estrategia los μ padres seleccionados generan a partir de operaciones de recombinación $\lambda >= \mu$ nuevos individuos, y entre la población total de $\boldsymbol{\mu+\lambda}$ se seleccionan los μ mejores individuos quienes serán los nuevos descendientes para la próxima generación del algoritmo, de esta forma de manera continua hasta satisfacer la condición de parada. (Luque del Arco-Calderón, Cristóbal., 2009).

1.3. LA ESTRATEGIA EVOLUTIVA: COMA (μ, λ)

Se parte de una población inicial de μ individuos, mediante el empleo de operaciones de recombinación se generan $\lambda >= \mu$ individuos a partir de los μ padres iniciales, de estos λ descendientes se escogen los μ mejores individuos quienes serán los nuevos individuos representantes para la próxima generación, todo esto de manera continua hasta satisfacer la condición de parada. (Murias, Rodríguez Ángel., 2007).

2. MATERIALES Y METODOS

2.1. CONSIDERACIONES PARA REALIZAR AUTO-ADAPTACIÓN DE MANERA ADECUADA

- μ > λ para poder disponer de diferentes estrategias de mutación (García, Carlos., García Edwin., Villada, Fernando., (2012).
- μ > λ para generar un excedente de hijos.
- Una presión selectiva alta, pero no demasiado, e.g., λ ≈ 7μ.
- Estrategia (μ, λ) para librarse de estrategias mal adaptadas.
- Realizar recombinación intermedia sobre los parámetros de la mutación.

2.2. ESTRUCTURA DEL ALGORITMO GENÉTICO PARA RESOLVER PROBLEMA DE MINIMIZACIÓN

El algoritmo genético está compuesto por las siguientes partes:

- **Representación del individuo**

Los individuos están compuestos de dos partes:

- Variables Objetivo: $(x_1, x_2, \dots, x_n)$
- Parámetros de la Mutación: $(\delta_1, \delta_2, \dots, \delta_n)$

Cada solución es por lo tanto: $[x_1, x_2, \dots, x_n , \delta_1, \delta_2, \dots, \delta_n]$

- **La Función Objetivo**

La función fitness (Luque del Arco-Calderón, Cristóbal., Isasi Viñuela Pedro., Hernández Castro Julio César., 2004), tiene como fin definir la sumatoria del cuadrado de las variables objetivos (problema de la Esfera), la cual representa nuestro objetivo a minimizar.

$$f(\vec{x}) = \sum_{i=1}^{n} {x_i}^2$$

- **Operador de Selección**

Típicamente, para las Estrategias Evolutivas, la selección de los padres para la reproducción se hace de manera aleatoria. Todas las soluciones tienen la misma probabilidad de ser seleccionada, y no se introduce sesgo en la búsqueda en esta fase. (Aguilar, J., & Rivas, F., 2001).

- **Operadores de Cruce**

Para este tipo de algoritmos con estrategias evolutivas y representación real de cromosomas con variables objetivo (Herrera, Francisco., Lozano Manuel., Sánchez, Ana M., 2002), la recombinación de estos puede hacerse de dos maneras: Discreta, elegir entre dos valores, o Intermedia, utilizando operadores de Cruce Aritméticos y Geométricos que promedien los valores, en los que dados dos padres: $\boldsymbol{Padre1} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n})$ y $\boldsymbol{Padre2} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_n})$ se crea un descendiente $\boldsymbol{Hijo} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \dots, \boldsymbol{z_n})$
como sigue:

- Para Recombinación Aritmética:

$$Z_i \frac{x_i + y_i}{2}$$

- Para Recombinación Geométrica:

$$z_i \sqrt{x_i * y_i}$$

- **Mutación Auto-Adaptativa**

Las variables se mutan añadiendo valores aleatorios que siguen una distribución normal. (Cantor, Giovanni., Gómez Jonatan., 2008). $\boldsymbol{x_i} = \boldsymbol{x_i} + \boldsymbol{N}\ (\boldsymbol{0}, \boldsymbol{\delta})$
La idea clave:

• δ es parte de la representación $x_1, \dots, x_n, \delta$

• δ también se mutará a δ’ .

El paso de la mutación δ’ co-evoluciona con la solución x.

- **Reemplazo**

El reemplazo se realiza de manera determinista, y las estrategias empleadas en este algoritmo de estudio son: Estrategia Coma (μ,λ) (1.2) y Estrategia Plus (μ + λ) (1.3). Donde la estrategia Plus es Elitista, y la estrategia Coma puede hacer perder la mejor solución, aunque suele preferirse para escapar de óptimos locales o seguir óptimos dinámicos, y menos insensible a súper-individuos.

- **Condición de Parada**

El algoritmo evolutivo detendrá su ejecución cuando su fitness cumpla con la condición de alcanzar un valor óptimo establecido, para el presente caso de estudio la condición es: $\boldsymbol{f(\vec{x}) \leq 10^{-6}}$

2.3. PARAMETRIZACIÓN

La configuración del algoritmo es la siguiente:

Tamaño de población de μ = 16 y un tamaño de la población descendente de λ = 100. Una condición de parada hasta que se encuentre una solución cuyo fitness cumpla $f(\vec{x}) \leq 10^{-6}$. Un operador de Selección aleatorio. Se establecieron dos operadores de estudio y comparación: Recombinación Discreta y Aritmética. Rango de Valores de Variables Objetivos [-10,10] y parámetros de mutación [0.000001,10]; En el Cuadro 1 se muestran los parámetros usados por el Algoritmo.

Cuadro 1: Parametrización (L = Longitud de Variables = (NxM) + 5)

Tamaño de la Población	**μ = 16**
Tamaño de la Población Descendente	**λ = 100**
Condición de Parada	$\boldsymbol{f(\vec{x}) \leq 10^{-6}}$
Operador de Selección	**Aleatoria.**
Operador de Cruce	**Recombinación Discreta y Aritmética, pc = 1/L**
Reemplazo	**Estrategia Coma (μ,λ) y Estrategia Plus (μ + λ)**

2.4. LA METODOLOGÍA APLICADA EN LOS EXPERIMENTOS

Una vez implementado el Algoritmo Genético en la herramienta informática MatLab, se desarrolló un Caso de Estudio en el que se comparan las dos variantes del algoritmo, tanto con la Estrategia de Reemplazo Coma (μ,λ) como con la Estrategia Plus (μ + λ), para conocer si existen diferencias significativas en el esfuerzo computacional necesario para lograr un valor objetivo $f(\vec{x}) \leq 10^{-6}$. Además se realizó un estudio adicional agregando un Operador de Recombinación Aritmético, para las Variables Objetivos, y de Recombinación Geométrico para los parámetros de mutación. Para el estudio se realizaron 10 ejecuciones independientes para resolver el problema con valores de n =10, 25 y 50, obteniendo de cada ejecución la mejor Traza encontrada, con información del comportamiento del algoritmo, así como el número de evaluaciones necesarias para cumplir con el objetivo establecido.

3. RESULTADOS Y DISCUSIÓN

3.1. Análisis del problema empleando el Operador de Recombinación Discreta

Después de correr el esquema básico de la metodología aplicada para el estudio (2.4) y con los parámetros mencionados en la sección (2.3), las Tablas 1 y 2 muestran los resultados obtenidos por las Estrategias evolutivas en estudio, los cuales representan el esfuerzo computacional requerido (medido en número evaluaciones) para lograr alcanzar un valor del fitness $f(\vec{x}) \leq 10^{-6}$, obtenido de 10 ejecuciones independientes para cada valor de n=10, 15 y 25. La figura 1 representa de forma gráfica, la media de estos resultados para cada valor de *N*.

Tabla 1: Esfuerzo Computacional, medido en número de evaluaciones, para lograr un $f(\vec{x}) \leq 10^{-6}$con la Estrategia Evolutiva Coma (μ,λ).

N	Ejecuciones										Mínimo	Mediana
	1	2	3	4	5	6	7	8	9	10		
10	11700	18400	21700	46400	14600	12500	20900	16100	27800	21900	11700	21200
25	46000	49400	46000	42400	43400	43100	46600	48900	42500	46500	42400	45480
50	104300	108900	103600	99700	105300	97800	94600	104600	102600	103200	94600	102460

Tabla 2: Esfuerzo Computacional, medido en número de evaluaciones, para lograr un $f(\vec{x}) \leq 10^{-6}$con la Estrategia Evolutiva Plus (μ + λ)

N	Ejecuciones										Mínimo	Mediana
	1	2	3	4	5	6	7	8	9	10		
10	30800	33200	18100	40300	31200	34400	34600	19100	37500	66700	18100	34590
25	35500	43200	41900	39400	35700	40200	44900	37300	42800	49000	35500	40990
50	32000	90900	92700	86500	92000	90800	90500	90000	93600	98200	86500	91720

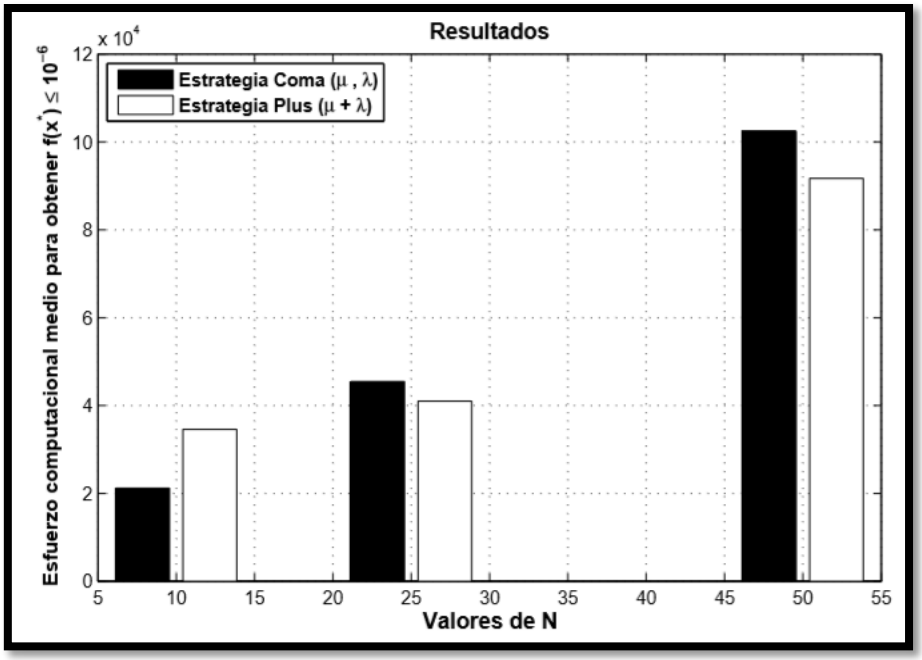

Figura 1: Esfuerzo computacional medio en 10 ejecuciones del algoritmo con Recombinación Discreta.

Sobre estos datos se aplicó el test estadístico de *Wilcoxon ranksum* (Hernández, Montiel L.A., Bonilla, Huerta E., Morales, Caporal L., 2011) para analizar comparativamente el rendimiento de las estrategias implementadas. En la tabla 3 se muestran los resultados de este test, donde la columna ***h*** nos indica si se debe o no rechazar la Hipótesis Nula, la cual indica que las medianas de los resultados son iguales, y la columna ***p*** que es el *p-value* retornado de este test, el cual es equivalente al U-test de *Mann-Whitney*.

Tabla 3: Test de Wilcoxon aplicado a los resultados generados por los algoritmos A y B (Tabla 1 y 2)

Test de Wilcoxon Rank sum		
Laberinto	**h**	**P**
10	0.01726	1
25	0.01722	1
50	0.00033	1

Entonces según el Test de Wilcoxon Ranksum, podemos concluir que los resultados de las dos Estrategias, para cualquier valor de n (10, 25, 50), tienen una diferencia estadísticamente significativa entre ellos, ya que nos indica que se debe rechazar la hipótesis Nula (con un nivel de significancia mayor al 5%) para los tres casos de estudio. Por lo que para obtener mejores soluciones, según el esfuerzo computacional, deberíamos

escoger una de las dos Estrategia de estudio, y según los resultados de la Tabla 1, en el caso que haya un valor de N=10 Variables Objetivo, deberíamos escoger **La Estrategia Coma** (μ,λ), ya que el Esfuerzo Medio en 10 ejecuciones del algoritmo es menor al valor obtenido con la Estrategia Plus (μ + λ). Para el caso de N=25 y 50, deberíamos escoger la **Estrategia Plus** (μ + λ), ya que de igual manera nos referencia de acuerdo al número medio de evaluaciones necesarias en 10 ejecuciones independientes.

3.2. Análisis del problema empleando el Operador de Recombinación Aritmética

Este mismo análisis y ejecución de resultados se aplicó pero con un Operador de Recombinación Aritmético aplicado a los nuevos individuos de la población descendente, cuyos resultados de ejecución se muestran en las Tabla 4 y 5, de igual forma se aplicó el Test de Wilcoxon Ranksum para conocer las diferencias significativas entre ellos.

Tabla 4: Esfuerzo Computacional, medido en número de evaluaciones, para lograr un $f(\vec{x}) \leq 10^{-6}$ con la Estrategia Evolutiva Coma (μ,λ).

N	Ejecuciones										Mínimo	Mediana
	1	2	3	4	5	6	7	8	9	10		
10	10600	10700	14800	12200	14600	8900	11600	8800	11300	14300	8800	11780
25	27800	29700	27200	26500	28600	29500	27600	25700	30200	26400	25700	27920
50	75100	78500	70100	78900	71900	73500	70700	71200	66900	73400	66900	73020

Tabla 5: Esfuerzo Computacional, medido en número de evaluaciones, para lograr un $f(\vec{x}) \leq 10^{-6}$ con la Estrategia Evolutiva Plus (μ + λ).

N	Ejecuciones										Mínimo	Mediana
	1	2	3	4	5	6	7	8	9	10		
10	8800	18300	8500	15200	9200	10000	12600	9500	8900	8700	8500	10970
25	26800	24200	28300	25000	25400	25200	25400	26700	24500	25000	24200	25650
50	60300	62700	63900	58500	59300	64700	57500	65700	63600	62900	57500	61910

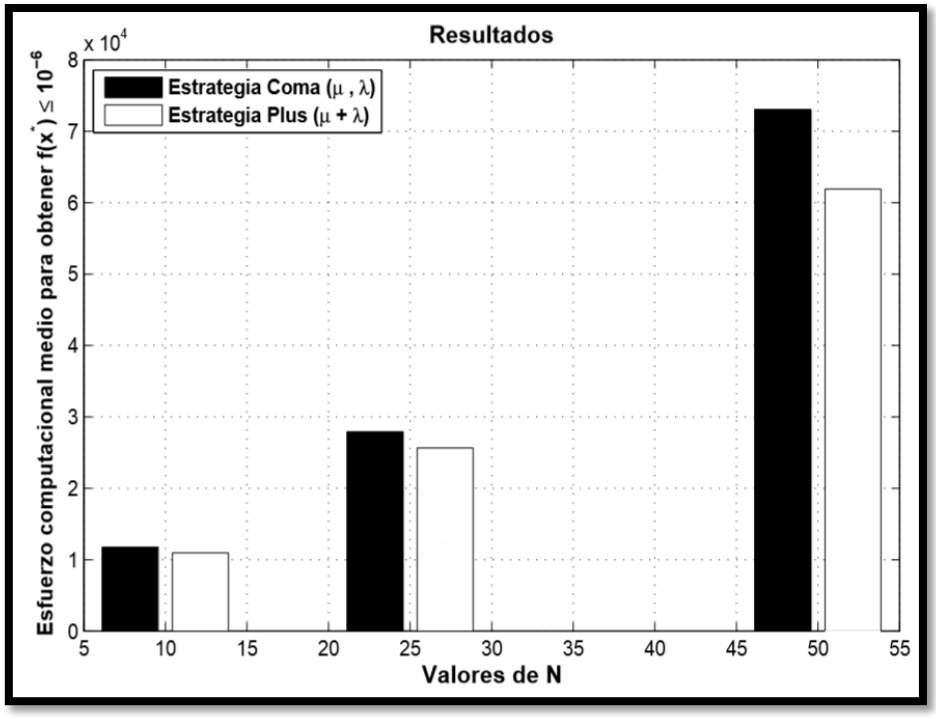

Figura 2: Esfuerzo Computacional Medio en 10 Ejecuciones del Algoritmo, con Recombinación Aritmética

Sobre estos datos se aplicó el test estadístico de *Wilcoxon ranksum* para analizar comparativamente el rendimiento de las estrategias implementadas. En la tabla 6 se muestran los resultados de este test

Tabla 6: Test de Wilcoxon Ranksum aplicado a los resultados generados por la Recombinación Aritmética (Tabla 4 y 5).

Test de Wilcoxon Rank sum		
Laberinto	**h**	**P**
10	0.27268	0
25	0.00456	1
50	0.00018	1

Según los resultados del Test para esta variante (Tabla 6), para el caso de N=25 y 50, existen diferencias en los resultados de las dos Estrategia en estudio, por lo que para obtener mejores soluciones deberíamos escoger la que mejor resultados nos genere, y según el esfuerzo computacional medio de 10 ejecuciones independientes, **La Estrategia Plus** (μ + λ) logra necesitar un menor número de evaluaciones para obtener soluciones según las condiciones requeridas.

En las figuras 3 y 4, se muestra el número de evaluaciones Mínimo para obtener $f(\vec{x}) \leq 10^{-6}$ en 10 ejecuciones independientes del algoritmo, así como la Traza con el comportamiento del Algoritmo cuya ejecución logró este menor esfuerzo computacional, tanto con el Operador de Recombinación Discreta y Aritmético.

4. CONCLUSIONES

En el presente trabajo hemos aplicado realizado un estudio experimental de optimización continuo empleado dos estrategias evolutivas para conocer su rendimiento basado en el esfuerzo computacional necesario para alcanzar un valor óptimo, resolviendo el problema de la Esfera con tres tamaño de variables N (10,25 y 50). Además se emplearon dos tipos de Recombinación dentro del algoritmo

Según los resultados obtenidos podemos concluir que empleado Recombinación Discreta, no existen diferencia significativas entre los dos; por lo que según el esfuerzo computacional, para N=10 la estrategia Coma converge más rápidamente al valor objetivo que la estrategia Plus, pero para las instancias de N=25 y 50 sucede lo contrario, la estrategia Plus converge más rápidamente que la estrategia Coma.

Además sobre el enfoque empleado con el tipo de Recombinación Aritmética, los resultados indican que para el caso de n=25 y 50, existen diferencias significativas en los resultados de las dos Estrategias, por lo que para obtener mejores soluciones deberíamos escoger la que mejor resultados nos genere, y según el esfuerzo computacional medio de 10 ejecuciones independientes, La Estrategia Plus (μ + λ) logra converger más rápidamente, Lo que finalmente nos permite concluir que esta estrategia es mucho más eficiente en emplearse en problemas de Optimización de mayor tamaño.

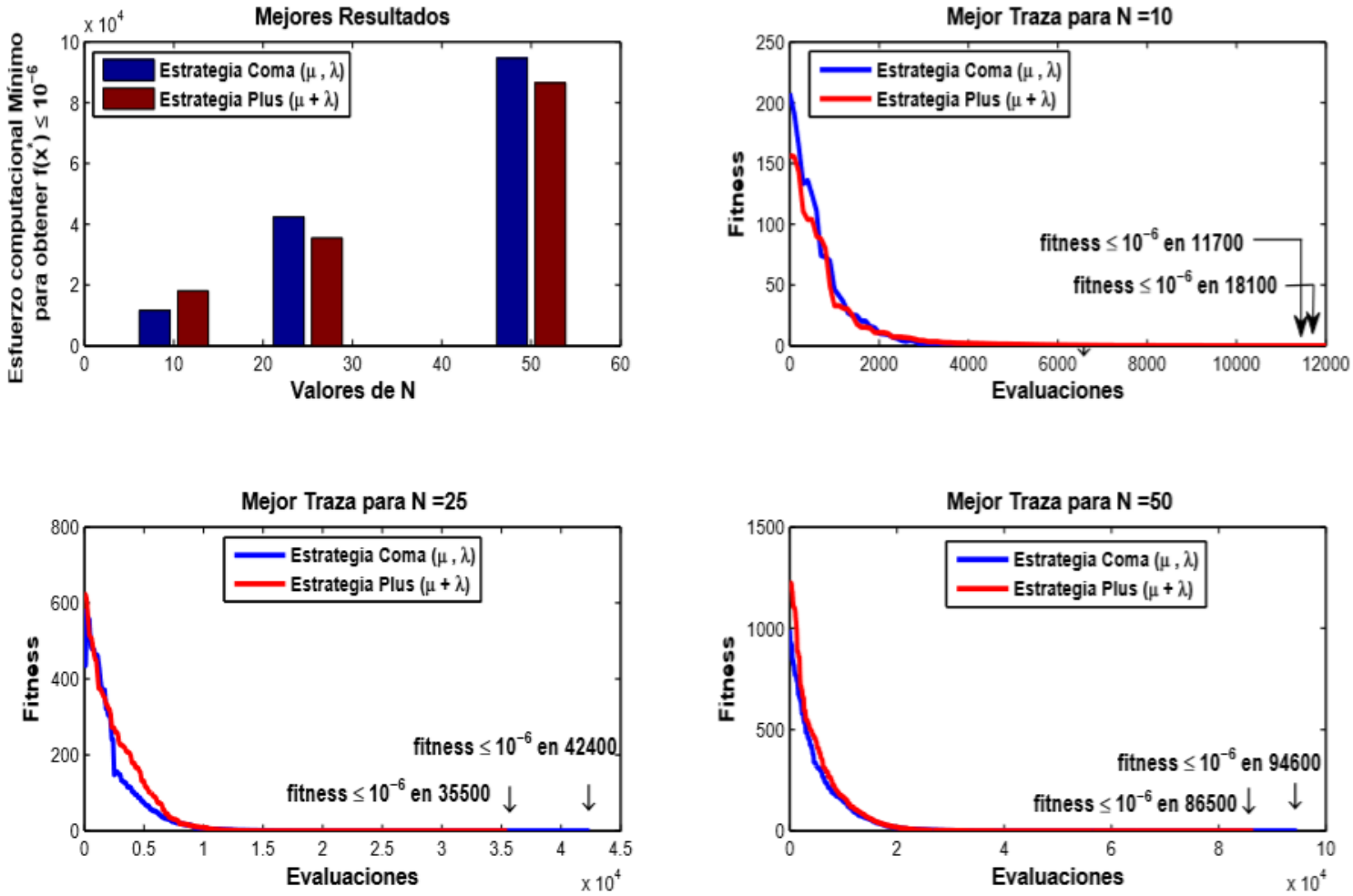

Figura 3: Mejor Traza en 10 Ejecuciones Independientes del Algoritmo con el Operador de Recombinación Discreta.

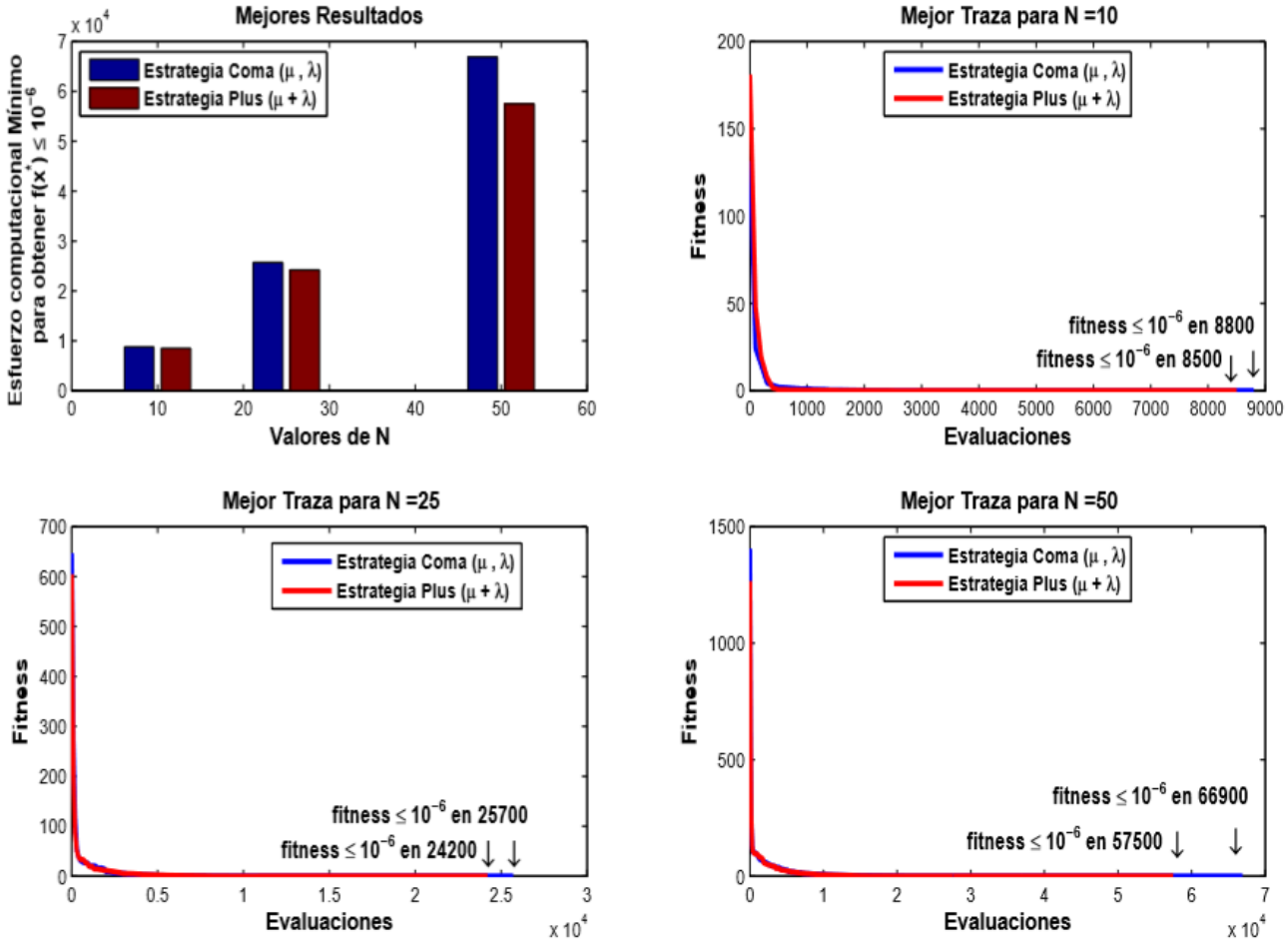

Figura 4: Mejor Traza en 10 Ejecuciones Independientes del Algoritmo con el Operador de Recombinación Aritmético.

5. REFERENCIAS BIBLIOGRÁFICAS

Aguilar, J., & Rivas, F. (2001)., Computación Inteligente., MERITEC, June.

Aguilar José, CEMISID., Facultad de Ingeniería Universidad de Mérida, Venezuela, Computación Inteligente (Computación Evolutiva).

Alba, Torres Enrique., (1999)., Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos., España.

Back, T., & Schwefel, H. P. (1996, May). Evolutionary computation: An overview. In Evolutionary Computation, 1996., Proceedings of IEEE International Conference on (pp. 20-29). IEEE.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–A comprehensive introduction. Natural computing, 1(1), 3-52.

Cantor, Giovanni., Gómez Jonatan., (2008)., ASIGNACIÓN DE PARÁMETROS EN LOS ALGORITMOS EVOLUTIVOS., Revista RE´TAKVN Facultad de Ingeniería - Universidad del Magdalena., Colombia.

Coello, Coello Carlos A., (2004)., Introducción a la Computación Evolutiva. México.

Estévez, Valencia Pablo., (1997)., OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS., Anales del Instituto de Ingenieros de Chile, Agosto 97, pp. 83-92.

García, Carlos., García Edwin., Villada, Fernando., (2012). Algoritmo Evolutivo Eficiente Aplicado a la Planeación de la Expansión de Sistemas de Distribución

García, S., Molina, D., Lozano, M., Herrera, F., Un estudio experimental sobre el uso de test no paramétricos para analizar el comportamiento de los algoritmos evolutivos en problemas de optimización, en Actas del Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, (MAEB07), 2007, pp.275-285.

Herrera, Francisco., Lozano Manuel., Sánchez, Ana M., (2002)., Operadores de Cruce con Múltiples Descendientes para Algoritmos Genéticos con Codificación Real: Estudio Experimental., Trabajo soportado por la RED HEUR TIC2002-10866-E y el ProyectoTIC2002-04036-C05-01

Hervás, C., & Ortiz, D. (2002). Operadores de cruce basados en estadísticos de localización para algoritmos genéticos con codificación real. In Primer Congreso Español De Algoritmos Evolutivos y Bioinspirados (AEB'02), Mérida, Spain (pp. 1-8).

López, Lara Adriana. (2003) . Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Departamento de Ingeniería Eléctrica, Sección de Computación, México, Un Estudio de las Estrategias Evolutivas para Problemas Multiobjetivo.

Luque del Arco-Calderón, Cristóbal., Isasi Viñuela Pedro., Hernández Castro Julio César.,(2004)., Distribución de Cargas en una Esfera Mediante Estrategias Evolutivas., IEEE LATIN AMERICA TRANSACTIONS, VOL. 2, NO. 2, JUNE 2004.

Moratilla, A., Fernández, E., Sánchez, J. J., & Vicario, B. (2014). Selección óptima de operadores para el tratamiento de problemas VRP con Algorítmos Genéticos. In Cuarta Conferencia Iberoamericana de Complejidad, Informática y Cibernética: CICIC.

Murias, Rodríguez Angel., (2007)., Estudio de bloques constructivos en algoritmos genéticos., España.

Salazar-Horning, E.J., Rojas-Oyarzún, R.S., (2010). Configuración multi–objetivo de sistemas de producción utilizando estrategias evolutivas. México.