

**DEFINICIÓN DE PESOS EN MOEA/D USANDO ARREGLOS DE CUBRIMIENTO
PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN DE MUCHOS
OBJETIVOS**



Universidad
del Cauca

CRISTIAN CAMILO ORDOÑEZ QUINTERO

Director: PhD. CARLOS ALBERTO COBOS LOZADA

Codirector interno: PhD. MARTHA ELIANA MENDOZA BECERRA

Codirector externo: PhD. HUGO ARMANDO ORDOÑEZ ERAZO

Asesor: PhD. JOSE TORRES JIMENEZ

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
MAESTRÍA EN COMPUTACIÓN
DEPARTAMENTO DE SISTEMAS
GRUPO DE I+D EN TECNOLOGÍAS DE LA INFORMACIÓN (GTI)
ÁREAS DE INVESTIGACIÓN: SISTEMAS INTELIGENTES

Popayán, octubre de 2019

CRISTIAN CAMILO ORDOÑEZ QUINTERO

**DEFINICIÓN DE PESOS EN MOEA/D USANDO ARREGLOS DE CUBRIMIENTO
PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN DE MUCHOS
OBJETIVOS**

Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de
la Universidad del Cauca para la obtención del Título de

Magíster en Computación

Director: PhD. CARLOS ALBERTO COBOS LOZADA

Codirector interno: PhD. MARTHA ELIANA MENDOZA BECERRA

Codirector externo: PhD. HUGO ARMANDO ORDOÑEZ ERAZO

(Universidad de San Buenaventura – Cali)

Asesor: PhD. JOSE TORRES JIMENEZ

(CINVESTAV - Centro de Investigación y de Estudios Avanzados del Instituto
Politécnico Nacional de México – Tamaulipas)

Popayán 2019

A Dios y su infinita sabiduría, por permitirme llegar hasta este momento tan importante de mi formación profesional.

A mis padres Jackeline Quintero, Oscar Daniilo Ordoñez por ser el pilar más importante en mi vida, por demostrarme siempre su cariño y apoyo incondicional.

A mis tíos Hugo Ordoñez y Mercedes Ordoñez por brindarme todo su apoyo, consejos en los momentos cuando más los necesito.

A mi novia Angélica Gaviria por su tiempo, dedicación y comprensión durante todo este tiempo.

A mis amigos Cristian F, Lina, Cristian M, Quienes fueron fundamentales en este proceso.

A Armando Ordoñez, quien me tendió su amistad, todos sus consejos y apoyo incondicional en el transcurso del tiempo

A mi tutor Carlos Cobos Lozada por brindarme su tiempo, paciencia, y consejos que en gran parte formaron mi carácter profesional y hacen lo que hoy en día soy.

TABLA DE CONTENIDO

CAPÍTULO 1.....	1
1 INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 APORTES	5
1.3 OBJETIVOS	5
1.3.1 OBJETIVO GENERAL.....	5
1.3.2 OBJETIVOS ESPECÍFICOS.....	5
1.4 RESULTADOS OBTENIDOS	6
1.5 ORGANIZACIÓN DEL DOCUMENTO	7
CAPÍTULO 2.....	9
2 CONTEXTO TEORICO Y ESTADO DEL ARTE.....	9
2.1 CONTEXTO TEORICO.....	9
2.1.1 OPTIMIZACIÓN DE MUCHOS OBJETIVOS	9
2.1.2 OPTIMIZACIÓN DE PARETO	10
2.1.3 FRENTE DE PARETO.....	11
2.1.4 PROBLEMAS Y MÉTRICAS DE EVALUACIÓN	12
2.1.5 MOEA FRAMEWORK	13
2.2 ESTADO DEL ARTE	14
2.2.1 INICIALIZACIÓN DE VECTORES DE PESOS EN MOEA/D	14
2.2.2 ALGORITMOS DE MUCHOS OBJETIVOS SOBRESALIENTES.....	16
2.2.3 NSGA-III	17
2.2.4 MOEA/D-DE	22
2.2.5 ARREGLOS DE COBERTURA	25
CAPÍTULO 3.....	27
3 PROPUESTA	27
3.1 MOEA/D-DE-ACA.....	27
3.2 COVERING ARRAYS Y AUGMENTED COVERING ARRAYS.....	28
3.3 HIPOTESIS PRINCIPAL.....	32
3.4 MOEA/D-DE-ACA.....	33
3.4.1 FUNDAMENTOS DE LA PROPUESTA.....	33

CAPÍTULO 4.....	36
4 EXPERIMENTACIÓN	36
4.1 PROBLEMAS USADOS PARA LA EVALUACIÓN Y COMPARACIÓN.....	36
4.2 MÉTRICA DE COMPARACIÓN.....	37
4.3 AJUSTE DE PARAMETROS	38
4.4 RESULTADOS DE MOEA/D-DE-ACA CON $v=9$ $t=2$	39
4.4.1 Test de Friedman para experimentos con $t=2$ y $v=9$	42
4.4.2 Tiempo de ejecución para experimentos con $t=2$ y $v=9$	43
4.4.3 Resultados por problema para experimentos con $t=2$ y $v=9$	44
4.1 EXPERIMENTOS CON FUERZA $t=2$ Y ALFABETO $v=17$	45
4.1.1 Test de Friedman para experimentos $t=2$ y $v=17$	47
4.1.2 Tiempo de ejecución para experimentos $t=2$ y $v=17$	49
4.1.3 Resultados por problema para experimentos $t=2$ y $v=17$	50
4.2 EXPERIMENTOS CON FUERZA 3 Y ALFABETO 9.....	51
4.2.1 Test de Friedman para experimentos $t=3$ y $v=9$	53
4.2.2 Tiempo de ejecución para experimentos $t=3$ y $v=9$	54
4.2.3 Resultados por problema para experimentos $t=3$ y $v=9$	56
CAPÍTULO 5.....	57
5 CONCLUSIONES Y TRABAJO FUTURO.....	57
CAPÍTULO 6.....	59
6 REFERENCIAS BIBLIOGRÁFICAS.....	59
Anexos	64

LISTA DE FIGURAS

Figura 1 Ejemplo de un problema de optimización con 3 objetivos (Adaptado de [1])	1
Figura 2 Frontera de Pareto de un problema de Minimización (Adaptado de López [29])	11
Figura 3 Espacio de búsqueda y espacio objetivo (adaptado de López [29])	11
Figura 4 Tipos de frente de Pareto.....	12
Figura 5 Ejemplo de la diferencia entre el funcionamiento de NSGA-II y NSGA-III	18
Figura 6 Distribución de 15 puntos de referencia para un problema con 3 objetivos tomado de [46].....	20
Figura 7 Calculo de distancia para asociar soluciones a los p de referencia 2D,3D Tomado de [46]	22
Figura 8 Problema de perdida de diversidad en MOEA/D	23
Figura 9 Actualización de soluciones MOEA/D-DE	23
Figura 10 Ejemplo CA fuerza 2 (Fuente propia)	26
Figura 11 Ejemplo CA fuerza 2	28
Figura 12 Cobertura PF de un CA.....	31
Figura 13 Cobertura PF de un ACA	31
Figura 14 relación vector de peso en espacio objetivo	34
Figura 15 Tiempo promedio de ejecución en segundos para todos los algoritmos en los experimentos con fuerza 2 y alfabeto 17.....	50
Figura 16 Valores de DGI obtenidos para todos los algoritmos en la evaluación de $v = 17$ y $t = 2$ de 10 a 100 objetivos. A: MOEA / D-DE-ACA, B: MOEA / D-DE y C: NSGA-III.....	51
Figura 17 Tiempo promedio de ejecución en segundos para todos los algoritmos en los experimentos con fuerza 3 y alfabeto 9	55
Figura 19 Valores de DGI obtenidos para todos los algoritmos en la evaluación de $v = 17$ y $t = 2$ de 10 a 100 objetivos. A: MOEA / D-DE-ACA, B: MOEA / D-DE y C: NSGA-III.....	56

LISTA DE ECUACIONES

Ecuación 1 Definición de un problema de optimización multi objetivo.....	9
Ecuación 2 Puntos de referencia Hiperplano	20
Ecuación 3 Puntos de referencia NSGA-III	21
Ecuación 4 Definición SOPs	27
Ecuación 5 Número de Soluciones CA	29
Ecuación 6 Número de soluciones Muestreadas	29
Ecuación 7 Distancia Generacional Invertida.....	38

LISTA DE TABLAS

Tabla 1 Métricas de Comparación	13
Tabla 2 CA (36;2,10,5)	30
Tabla 3 ACA (42;2,10,5).....	32
Tabla 4 Propiedades de los problemas DTLZ y WFG usados para la evaluación donde $M = \{10, 20, 30, \dots, 100\}$	37
Tabla 5 Ajustes de parámetros para los algoritmos comparados.....	38
Tabla 6 Resultados de evaluación en problemas DTLZ con ACA $V=9$ $T=2$	40
Tabla 7 Resultados de evaluación en problemas WFG con ACA $V=9$ $T=2$	41
Tabla 8 Resultados test de Friedman y post hoc de Holm en DGI para fuerza ($T=2$) y alfabeto ($V=9$)	43
Tabla 9 Resultados de evaluación en problemas DTLZ con ACA $V=17$ $T=2$	47
Tabla 10 Resultados de evaluación en problemas WFG con ACA $V=17$ $T=2$	48
Tabla 11 Resultados test de Friedman y post hoc de Holm en DGI para fuerza ($T=2$) y alfabeto ($V=17$)	49
Tabla 12 Resultados de evaluación en problemas DTLZ con ACA $V=9$ $T=3$	52
Tabla 13 Resultados de evaluación en problemas WFG con ACA $V=9$ $T=3$	53
Tabla 14 Resultados test de Friedman y post hoc de Holm en DGI para fuerza ($T=3$) y alfabeto ($V=9$)	54

LISTA DE SIGLAS

CA	Covering Arrays / Arreglos de Cubrimiento
ACA	Augmented Covering Arrays / Arreglos de Cubrimiento Aumentados
MOEA/D	Multi Objective Evolutionary Algorithm based on Decomposition / Algoritmo Evolutivo Multi Objetivo basado en Descomposición
MOEA/D-DE	Multi Objective Evolutionary Algorithm based on Decomposition, Differential Evolution / Algoritmo Evolutivo Multi Objetivo basado en Descomposición con Evolución Diferencial
MOEA/D-DE-ACA	Multi Objective Evolutionary Algorithm based on Decomposition, Differential Evolution and Augmented Covering Arrays / Algoritmo Evolutivo Multi Objetivo basado en Descomposición, Evolución Diferencial y Arreglos de Cubrimiento Aumentados
NSGA- III	Non-dominated Sorting Genetic Algorithm III / Algoritmo Genético de Ordenamiento No Dominado III
PF	Pareto Front / Frente de Pareto
MOEAs	Multi Objective Evolutionary Algorithms / Algoritmos Evolutivos Multi Objetivos
MaOEAs	Many Objective Evolutionary Algorithms / Algoritmos Evolutivos de Muchos Objetivos
MOPs	Multi Objective Optimization Problems / Problemas de Optimización Multi Objetivo
MaOPs	Many Objective Optimization Problems / Problemas de Optimización de Muchos Objetivos
DGI	Distancia generacional Invertida
SOPs	Scalar Optimization Problems / Problemas de optimización escalares

LISTA DE ANEXOS

Anexo 1 Artículo I: Análisis comparativo de MOGBHS con otros algoritmos de última generación para problemas de optimización de objetivos múltiples.....	64
Anexo 2 Artículo II: Definición de vectores de peso en MOEA / D-DE utilizando matrices de cobertura aumentada para la optimización de muchos objetivos	76
Anexo 3 Artículo III: Mejora de K-means usando arreglos de cobertura para la agrupación de procesos empresariales.....	96

ABSTRACT

Abstract— Many-objective optimization problems are today ever more common. Among the evolutionary algorithms used in their solution, the decomposition-based approach stands out, with MOEA/D and its variations playing major roles. MOEA/D variations seek to improve the definition of weight vectors and evolution operators - selection, crossover, mutation, and replacement. Although the success of the algorithm depends heavily on how well the weight vectors are defined in decomposing the problem, research on the topic is scarce. To better define weight vectors, this paper proposes the use of augmented covering arrays (ACAs), mathematical objects that enable a better sampling of interactions of M objectives using the least number of weight vectors, based on an interaction level, or strength, previously defined by the user. Comparing the proposed method in DTLZ and WFG problems versus traditional weight vectors definition in MOEA/D-DE and results obtained against NSGA-III, it was seen that in small to medium populations (up to 850 solutions) with 40 to 100 objectives, better results are obtained in Inverted Generational Distance supported by Friedman nonparametric and Holm post-hoc tests, with significant reductions in execution time. The use of ACAs enables weight vectors definition in nonlinear (logarithmic) time as the number of objectives grows. MOEA/D variations and decomposition-based algorithms may thus be expected to include the proposed approach and thereby improve on results in the literature.

Key words: optimization methods, optimization of many objectives, decomposition, differential evolution.

RESUMEN

Los problemas de optimización de muchos objetivos son cada día más comunes. Entre los algoritmos evolutivos utilizados para su solución, destaca el enfoque basado en descomposición con el algoritmo MOEA/D y sus variaciones. Las variaciones de MOEA/D buscan mejorar la definición de vectores de peso, la adaptación de estos en el proceso de búsqueda y la búsqueda en sí misma, modificando los operadores de selección, cruce, mutación y reemplazo, entre otros. Aunque el éxito del algoritmo depende en gran medida de qué tan bien se definan los vectores de peso al descomponer el problema, la investigación sobre el tema es escasa. Para definir los vectores de peso, en esta tesis se propone el uso de arreglos de cobertura aumentada (ACA), unos objetos matemáticos que permiten un mejor muestreo de las interacciones de los M objetivos, utilizando el menor número de vectores de peso, en función de un nivel de interacción o fuerza previamente definido por el usuario. Al comparar el método propuesto en problemas de las familias DTLZ y WFG versus la definición de vectores de peso en MOEA/D-DE (una variante de MOEA/D que usa los operadores de evolución diferencial) y los resultados obtenidos contra NSGA-III, se observó que en poblaciones pequeñas a medianas (hasta 850 soluciones) con 40 a 100 objetivos, se mejoran los resultados de los algoritmos medidos en distancia generacional invertida, lo cual se respalda con los resultados de la prueba estadística no paramétrica de Friedman y el post hoc de Holm, con reducciones significativas en el tiempo de ejecución. El uso de ACAs permite la definición de vectores de peso en tiempo no lineal (logarítmico) a medida que crece el número de objetivos. Con lo anterior, se espera que las variaciones de MOEA/D y de otros los algoritmos basados en descomposición incluyan el enfoque propuesto y, por lo tanto, mejoren los resultados reportados a la fecha en la literatura.

Palabras clave: Métodos de Optimización, Optimización de muchos objetivos, descomposición, evolución diferencial.

CAPÍTULO 1

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad el ser humano dentro de su cotidianidad se enfrenta a diferentes desafíos y problemas donde debe tomar decisiones que en muchos casos son complejas. A continuación, se muestra un ejemplo sencillo donde una persona desea comprar un automóvil entre cuatro alternativas posibles, a saber: Ford, Toyota, Honda y Mazda (ver **Figura 1**).

3 OBJETIVOS

4 ALTERNATIVAS	AUTOMOVIL	VALOR COMERCIAL	CONSUMO DE GASOLINA (G/100 KM)	POTENCIA (P)
	FORD	29300000	9	252 (P)
	TOYOTA	25000000	8.2	132 (P)
	HONDA	26500000	8.5	150 (P)
	MAZDA	24000000	11	167 (P)

Figura 1 Ejemplo de un problema de optimización con 3 objetivos (Adaptado de [1])

La decisión de comprar uno de los cuatro automóviles del ejemplo, contempla tres características a saber: precio (valor comercial), consumo de gasolina y potencia. Estas características se convierten en objetivos para el comprador, buscando por ejemplo un carro con un valor comercial económico (*minimizar el precio*), que consuma poca gasolina (*minimizar consumo de gasolina*) y además que sea el más potente (*maximizar potencia*).

¿Qué carro se debería comprar con base en estos 3 objetivos y los datos de la **Figura 1**? Es evidente que tomar una decisión no es fácil, Si se escoge el Mazda que es el más económico se toma el que consume más gasolina, lo que no se desea, pero el segundo en mayor potencia. Si se selecciona el Toyota se escoge el segundo carro más económico en precio y el que menos consume gasolina (que es muy bueno) pero el de menor potencia.

Los MOEAs (Multi-Objective Evolutionary Algorithms) [2] son una herramienta útil para resolver este tipo de problemas, ya que permiten encontrar un conjunto de soluciones (no una única solución), a problemas con baja cantidad de objetivos (hasta 3), denominados MOPs (Multi-objective Optimization Problems), objetivos que pueden estar en conflicto. En la última década, se han propuesto diferentes algoritmos para abordar este tipo de problemas, pudiendo clasificar a las propuestas más exitosas en tres enfoques principales: basados en dominancia, basados en

indicadores y basados en descomposición [3]. Entre los principales algoritmos se pueden destacar a NSGA-II [4], SPEA2 [5], MOEA/D [6] y MSOPS [7]. Estos algoritmos se usan comúnmente para optimizar sistemas con baja cantidad de objetivos (hasta 3), como por ejemplo, la planificación de rutas aéreas [8], el diseño de acueductos y alcantarillados [9], la optimización de rutas de un sistema de transporte público terrestre [10], entre otros. Sin embargo, cuando se trata de problemas de muchos objetivos (Many-Objective Optimization Problems, MaOPs), es decir, problemas con cuatro (4) o más objetivos, los MOEAs tradicionales son propensos a fallar o a converger a óptimos locales debido a que entre otras cosas, con muchos objetivos se hace difícil definir cuando una solución domina a otra [11].

En los últimos años se han propuesto diversos algoritmos evolutivos para optimización de muchos objetivos (Many-Objective Evolutionary Algorithms, MaOEAs) que buscan superar las deficiencias de los tradicionales MOEAs. Estos algoritmos también tienen diversos enfoques, entre los más importantes están [11]:

- **Basados en escalarización de funciones (descomposición/agregación):** Los basados en descomposición buscan resolver el problema descomponiéndolo en varios problemas mono objetivo, usando múltiples funciones ponderadas de los objetivos donde cada objetivo tiene diferentes valores de peso en cada función. Los basados en agregación, usan funciones para combinar grupos de objetivos y de esta forma trabajar con una cantidad mucho menor de éstos y resolverlos con MOEAs tradicionales. De este enfoque se destaca el algoritmo MOEA/D [6] y variaciones de éste que incluyen por ejemplo la mejora de los operadores de evolución basado en evolución diferencial como en MOEA/D-DE [12] y en MOEA/D-oDE [13].
- **Basados en conjuntos de referencia:** se basan en una lista de soluciones que se encuentran en un conjunto de referencia, un conjunto que orienta el proceso de búsqueda. En esta categoría se destaca el algoritmo de dos archivos mejorado TAA [14] y en especial, la versión 3 del algoritmo de clasificación no dominada NSGA-III [15].
- **Basados en indicadores de calidad:** Estos algoritmos transforman el problema de muchos objetivos a un problema de optimización de un sólo objetivo, éste último, representa que tan buenas son las soluciones frente a las demás de la población en el espacio de objetivos (indicador). De estos los más reconocidos son IBEA [16], I-SIBEA [17] y el algoritmo de colonia de abejas artificiales E-MOABC [18].
- **Basados en reducción dimensional:** algoritmos que toman los muchos objetivos del problema original y los reducen en una representación de baja dimensionalidad usando técnicas como el Análisis de Componentes Principales

(Principal Component Analysis, PCA), selección no supervisada de características y técnicas greedy. Se puede destacar en este enfoque a PCA-NSGA-II [19] entre otros.

- **Basados en el particionamiento del espacio:** Algoritmos que en cada iteración del proceso evolutivo sólo optimizan subconjuntos de objetivos del problema, destacándose en esta categoría el algoritmo ϵ R-EMO [20].

El enfoque basado en descomposición ha atraído mucho la atención de los investigadores del área, en especial el algoritmo MOEA/D [6] que ha sido sujeto de diversas mejoras que buscan entre otras: 1) el desarrollo de nuevos métodos de definición de los vectores de pesos (ponderación) que descomponen el problema de muchos objetivos a múltiples problemas mono objetivo, 2) el uso de nuevos enfoques de descomposición, 3) la eficiente asignación de recursos computacionales, 4) la mejora en el proceso de búsqueda, modificando los operadores de selección, cruce, mutación y reemplazo del algoritmo, y 5) la hibridación con enfoques basados en dominancia, entre otros.

Una de las limitaciones de MOEA/D que ha sido menos investigada se centra en la definición de los vectores de pesos que descomponen el problema. En su versión original el algoritmo hace un muestreo uniformemente aleatorio de los pesos de los distintos objetivos, la cual no garantiza el muestreo adecuado de la interrelación entre los objetivos o genera un incremento exponencial en la cantidad de vectores requeridos, en la medida en que el número de objetivos crece, además de un mayor tiempo de ejecución [21]. Otra opción es la basada en el método simplex [22] donde el tamaño de la población (el número de vectores de peso) aumenta de forma no lineal con el número creciente de objetivos. Aquí el tamaño de la población no lo puede definir el usuario y la distribución de los vectores de peso no garantiza que las soluciones muestreen la interacción entre los objetivos [23].

En UMOEA/D se adopta el método de diseño uniforme para establecer los vectores de pesos de los subproblemas, además evalúan la ventaja de su diseño frente a la versión original de MOEA/D y NSGA-II. En comparación con MOEA/D, la distribución de los vectores de pesos es más uniforme en el espacio de objetivos y el tamaño de la población aumenta de forma no lineal con el número de objetivos [23].

Recientemente en MaOEA/D-2ADV se ha realizado el ajuste dinámico del número de vectores de pesos que se usan para orientar el proceso de búsqueda combinado con el análisis del Frente de Pareto obteniendo resultados superiores en problemas de muchos objetivos [24].

Teniendo en cuenta que la definición de los vectores de peso todavía es un aspecto que se debe mejorar en MOEA/D, en esta investigación se propuso el uso de

Arreglos de Cubrimiento (Covering Arrays, CAs), en especial un nuevo tipo de CAs denominados Arreglos de Cubrimiento Aumentados (Augmented Covering Arrays, ACAs) para la definición de la ponderación de pesos, teniendo en cuenta que este objeto matemático garantiza el mayor cubrimiento (mayor muestreo de la interacción de diferentes factores, en este caso objetivos de optimización) con el menor esfuerzo posible [25]. Los CAs en general, han sido utilizados para soportar el diseño experimental (en campos como la agricultura, la medicina, la biología, el diseño de materiales, entre otros) y últimamente son una de las herramientas más utilizadas para las pruebas de software y hardware. En todos estos campos, se requiere probar combinaciones de distintos factores sin realizar una búsqueda exhaustiva de los mismos debido a restricciones de costo, tiempo y esfuerzo.

Un arreglo de cubrimiento se define formalmente como CA (N, k, v, t) y se expresa como una matriz numérica de N filas y k columnas (factores), donde cada fila del CA representa una prueba/experimento que indica una combinación de valores v (alfabeto de los factores) de los k factores con una cobertura determinada por el parámetro t denominado fuerza. La cobertura basada en t es una medida del cubrimiento de las interacciones entre los diferentes factores, de manera que entre mayor sea la cobertura, más pruebas se requieren y cuando dicha cobertura (fuerza t) es igual al número de factores k , el arreglo equivale a una prueba exhaustiva.

Al aplicar CAs en diferentes campos, los investigadores han observado entre otros: 1) el crecimiento en el número de factores hace crecer logarítmicamente el número de filas del CA, lo que permite mantener controlados el costo, tiempo y esfuerzo, y 2) dependiendo del valor de t , las combinaciones que se obtienen en un CA implícitamente evalúan la interacción entre los factores. Estas dos observaciones son muy relevantes para la optimización de muchos objetivos en MOEA/D ya que permite definir el número de vectores a un valor que sólo crece logarítmicamente con el número de objetivos y además permite definir según un alfabeto específico los valores de peso de dichos vectores [26]. En el presente trabajo se propone el uso de ACAs (diseñados en el marco de esta investigación) en lugar de CA tradicionales, ya que estos permiten hacer un mejor muestreo de las ecuaciones diofánticas que se presentan al definir los vectores de peso en MOEA/D.

Por lo anterior, en esta investigación se responde a la siguiente pregunta de investigación: ¿Cuál es el efecto de incorporar el uso de Augmented Covering Arrays (ACAs) como soporte del proceso de inicialización de los vectores de peso del algoritmo MOEA/D-DE en la resolución de problemas de optimización de muchos objetivos (4 o más objetivos)?

Esta pregunta de investigación se evalúa en el contexto de problemas reconocidos por el estado del arte y disponibles en MOEA-framework, a saber, problemas de las

familias DTLZ y WFG, usando una medida clásica para evaluar la calidad como lo es la Distancia Generacional Invertida (DGI).

1.2 APORTES

Los aportes de este trabajo se definen desde tres perspectivas:

- **Investigación:** Se contribuye a la generación de nuevo conocimiento para la comunidad académica y científica internacional de sistemas inteligentes, a través de la propuesta de una nueva versión de un algoritmo evolutivo de optimización de muchos objetivos que se basa en MOEA/D-DE y la inicialización de sus vectores de pesos basado en arreglos de cubrimiento, denominado MOEA/D-DE-ACA. En la actualidad no se ha encontrado ningún reporte de investigación con el mismo enfoque en las bases de datos de ACM, IEEE, ScienceDirect, ni en Springer Link.
- **Innovación y desarrollo:** El algoritmo se implementó en Java dentro de MOEA Framework, conservando la arquitectura del framework y permitiendo la evaluación con problemas de las familias DTLZ y WFG ya disponibles en el framework, así como el uso de DGI como medida de evaluación de la calidad de las soluciones. Algunas pruebas incluyeron el uso del Hypervolumen, pero esta medida de calidad se desechó porque es muy costosa en tiempo de ejecución.
- **Transferencia:** Al contar con el algoritmo implementado en el MOEA Framework, el uso de este en problemas científicos, industriales y empresariales se facilitará, ya que este framework ya es usado por mucha gente alrededor del mundo. Además, la publicación de los resultados servirá como mecanismo de difusión del algoritmo, sus resultados y su uso libre en el framework.

1.3 OBJETIVOS

A continuación, se presentan los objetivos tal como fueron aprobados por parte del Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

1.3.1 OBJETIVO GENERAL

Definir el proceso de inicialización de los vectores de ponderación de pesos en MOEA/D basado en arreglos de cubrimiento para resolver problemas de optimización de muchos objetivos.

1.3.2 OBJETIVOS ESPECÍFICOS

- Definir la línea base de comparación del trabajo investigativo, compuesta por dos de los algoritmos más competitivos del estado del arte en la optimización de

muchos objetivos y los problemas y medidas de calidad aceptados por la comunidad académica y científica integrados en MOEA Framework.

- Crear una nueva versión del algoritmo MOEA/D para resolver problemas de optimización de muchos objetivos modificando el proceso de inicialización de los vectores de ponderación de pesos usando arreglos de cubrimiento.
- Evaluar la calidad de la nueva versión del algoritmo propuesto a través de medidas clásicas del área (por ejemplo, distancia generacional invertida) y el uso de problemas de las familias DTLZ y WFG principalmente, así como, comparar los resultados con dos métodos recientemente planteados en el estado del arte y su análisis mediante test estadísticas no paramétricas de Friedman y Wilcoxon.

1.4 RESULTADOS OBTENIDOS

A continuación, se presenta un resumen de los resultados más destacados que se obtuvieron en este trabajo de grado:

- **Monografía de trabajo de grado:** Se refiere al presente documento el cual presenta el estado del arte del problema y realiza una descripción de los algoritmos utilizados. Presenta la integración de ACAs en el proceso de inicialización de MOEA/D-DE dentro del framework MOEA. Por último, expone los resultados obtenidos de la evaluación y comparación de la propuesta frente a dos algoritmos, junto con las conclusiones y el trabajo futuro que se espera realizar en el área.
- **Aplicación software:** Módulo en MOEA-framework, escrito en lenguaje de programación Java que permite realizar la experimentación con la nueva versión de MOEA/D-DE, que usa ACAs para la inicialización de los vectores de peso.
- **Artículo 1:** Artículo presentado y publicado en MICAI 2017 - 16th Mexican International Conference on Artificial Intelligence, Ensenada, México, October 23–28 de 2017. Con la siguiente referencia: Ordoñez, C., Ruano, E., Cobos, C., Ordoñez, H., Ordoñez, A. Comparative Analysis of MOGBHS with Other State-of-the-Art Algorithms for Multi-objective Optimization Problems (2018). Lecture Notes in Computer Science (LNCS), vol. 10632. 154-170. ISSN: 0302-9743. https://doi.org/10.1007/978-3-030-02837-4_13. La revista LNCS se encuentra indexada en SJR con categoría Q2 para 2018 año de publicación del artículo, ver **Anexo 1**.
- **Artículo 2:** Artículo en proceso de evaluación en revista internacional con la siguiente referencia: Ordoñez, C., Cobos, C., Torres-Jiménez, J., Ordoñez, H., Mendoza, M. Weight Vectors Definition in MOEA/D-DE using Augmented Covering Arrays for Many-Objective Optimization. IEEE Access. On evaluation, pp. 1-20. ISSN: 1089-778X. Esta revista esta indexada en JCR con categoría Q1, ver **Anexo 2**.

- **Artículo 3:** Con el objetivo de explorar el uso de los arreglos de cobertura en otras áreas, en específico el clustering de procesos de negocio, se realizó una colaboración con la Universidad de San Buenaventura y se logró la aceptación para publicación de un artículo en CSEI'2019 - I Congreso de Ciencia de la Computación, Electrónica e Industrial, a realizarse del 28 al 31 de octubre de 2019 en Ibarra, Ecuador. Este artículo tiene la siguiente referencia: Ordoñez, H., Ordoñez, C., Cobos, C., Torres-Jiménez, J., Bucheli, V. Mejora de K-means usando arreglos de cubrimiento incrementales para la agrupación de procesos empresariales (2019). RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação. In Press, pp. 1-20. ISSN: 1646-9895. Esta revista está indexada en SJR con categoría Q3, ver **Anexo 3**.

1.5 ORGANIZACIÓN DEL DOCUMENTO

A continuación, se describe de manera general el contenido y organización de la presente monografía:

CAPÍTULO 1: INTRODUCCIÓN. Se refiere al presente capítulo que introduce el tema de investigación, presenta la pregunta de investigación, los aportes obtenidos en la investigación, los objetivos (general y específicos) definidos en el anteproyecto, un breve resumen de los resultados obtenidos y finalmente la organización de la monografía.

CAPÍTULO 2: CONTEXTO TEÓRICO Y ESTADO DEL ARTE. Este capítulo presenta en forma resumida una descripción teórica del tema de investigación luego describe el enfoque de descomposición y por último presenta las investigaciones más relevantes desarrolladas para resolver el problema de optimización de muchos objetivos, organizándolas por categorías según el enfoque de solución.

CAPÍTULO 3: ALGORITMO PROPUESTO: En este capítulo se describe el algoritmo propuesto denominado MOEA/D-DE-ACA, como también los arreglos de cubrimiento aumentados (Augmented Covering Arrays, ACAs) propuestos para cada uno de los experimentos llegando hasta 100 objetivos.

CAPÍTULO 4: EXPERIMENTACIÓN: Este capítulo presenta un resumen del proceso de investigación llevado a cabo en el transcurso de la tesis, además las métricas y problemas utilizados como también el afinamiento de parámetros de las meta-heurísticas y las pruebas realizadas para seleccionar el mejor conjunto de valores para su ejecución. En este capítulo se presentan los distintos resultados de la experimentación (distancia generacional invertida y tiempo de ejecución) en 3 escenarios, además de los resultados de la prueba estadística no paramétrica de Friedman con el post test de Holm y la discusión de dichos resultados.

CAPÍTULO 5: CONCLUSIONES Y TRABAJO FUTURO: En este capítulo se presentan las conclusiones obtenidas al finalizar el trabajo de grado e ideas que el grupo de investigación espera realizar en el futuro cercano.

CAPÍTULO 6: BIBLIOGRAFIA: Este último capítulo contiene las referencias bibliográficas de los artículos, libros y demás recursos consultados para la realización del proyecto.

CAPÍTULO 2

2 CONTEXTO TEORICO Y ESTADO DEL ARTE

2.1 CONTEXTO TEORICO

2.1.1 OPTIMIZACIÓN DE MUCHOS OBJETIVOS

Ante de definir un problema de optimización de muchos objetivos, primero se debe definir un problema de optimización multi objetivo (MOP), el cual puede ser definido con base en la **Ecuación 1**. Donde $x = (x_1, \dots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n$; $y = (y_1, \dots, y_k) \in \mathcal{Y} \subseteq \mathbb{R}^k$, x es una variable de decisión vectorial n -dimensional, y es el vector objetivo k -dimensional, $\mathcal{X} \subseteq \mathbb{R}^n$ expresa el espacio de decisión, $\mathcal{Y} \subseteq \mathbb{R}^k$ expresa el espacio objetivo. El conjunto de restricciones dadas por la segunda línea de la ecuación define la región de factibilidad $\mathcal{X}_f \subseteq \mathbb{R}^n$ y cualquier punto $x \in \mathcal{X}_f$ es una solución factible [2].

$$\text{Min } (y = F(x) = (f_1(x), f_2(x), f_3(x), \dots, f_k(x)))$$

$$\text{Sujeto a: } g(x) = (g_1(x), \dots, g_m(x)) \leq 0$$

Ecuación 1 Definición de un problema de optimización multi objetivo

El objetivo de la optimización multiobjetivo es aproximar las soluciones al Frente de Pareto (Pareto Front, PF) en el espacio de los objetivos de manera que ninguna mejora adicional en cualquier objetivo puede alcanzarse sin dañar al resto de los objetivos. Cuando se trata de optimizar cuatro (4) o más objetivos, a esta área de optimización se le denomina optimización de muchos objetivos. Un problema de optimización de muchos-objetivos (Many-objective Optimization Problems, MaOPs) se define como un MOPs con 4 o más objetivos [27]. Esta área nace debido al creciente número de problemas con muchos objetivos donde los algoritmos tradicionales son propensos a fallar o a converger de manera prematura a óptimos locales. A continuación, se detallan algunas situaciones que se presentan con los MOEAS para resolver MaOPs:

1. **Deterioro del PF.** Cuando se tiene un amplio número de objetivos, los algoritmos no pueden determinar las mejores soluciones, ya que al comparar dos soluciones en unos objetivos gana una y en los otros gana la otra, esto hace que los MOEAs converjan de manera anticipada a óptimos locales del PF.
2. **Dificultad para seleccionar y reemplazar soluciones.** Al no poder determinar cuál solución es mejor que otra, el proceso evolutivo o iterativo de mejora se dificulta, ya que la población se llena de soluciones no dominadas, esto debilita la calidad de la selección de soluciones basado en dominancia de Pareto y

cuando se crean nuevas soluciones se dificulta definir a cuál remplazar en la población, ya que ninguna domina a otra.

- 3. Aumento exponencial de soluciones para aproximar el frente de Pareto.** Al tener muchos objetivos, el frente de Pareto requiere de muchas más soluciones (el cual crece exponencialmente con el número de objetivos) para tener una representación más aproximada, además, se hace mucho más compleja la visualización de estas soluciones [11].

En los últimos años se han propuesto diversos algoritmos evolutivos para optimización de muchos objetivos que buscan superar las deficiencias de los tradicionales MOEAs [28]. Estos algoritmos incorporan diversos enfoques. En esta investigación se selecciona el enfoque de descomposición debido a que ha mostrado resultados muy competitivos y ha atraído la atención de los investigadores del área, en especial el algoritmo MOEA/D [6] que ha sido sujeto de diversas mejoras las cuales buscan combatir la convergencia prematura a soluciones óptimas locales en problemas de muchos objetivos.

2.1.2 OPTIMIZACIÓN DE PARETO

El concepto del óptimo de Pareto se presenta gráficamente en la **Figura 2**, donde la solución “e” pertenece al conjunto de soluciones óptimas (o soluciones no dominadas) dado que no puede encontrarse una solución “a” tal que mejore uno de los objetivos sin empeorar al menos uno de los otros. En este caso, la solución “g” es dominada por “e” y por “f”.

El conjunto de Pareto se define entonces como el conjunto de puntos en el espacio de búsqueda que se corresponden con las mejores soluciones al problema de optimización; y la Frontera de Pareto, como el conjunto de soluciones óptimas en el espacio objetivo. Este conjunto de soluciones se integra con aquellos puntos encontrados en el espacio objetivo que se corresponden con las soluciones no dominadas, de acuerdo con el criterio de dominancia de Pareto. El concepto de conjunto de Pareto y frontera de Pareto se representan en la **Figura 3**.

La frontera de Pareto obtenida puede ser lineal, cóncava, convexa, continua o discontinua dependiendo de las funciones objetivo integrantes del problema. Todas las soluciones pertenecientes a la frontera son igualmente buenas, y no se puede especificar si alguna de las soluciones es preferida sobre las otras, excepto en aquellos casos en que se haya definido una preferencia a priori.

En la optimización por métodos de Pareto, el desafío consiste en encontrar la Frontera de Pareto real, o lograr la mejor aproximación de esta sujeta a las limitaciones de recursos (tiempo y memoria) existentes. Por esta razón, el proceso de búsqueda utilizado es decisivo.

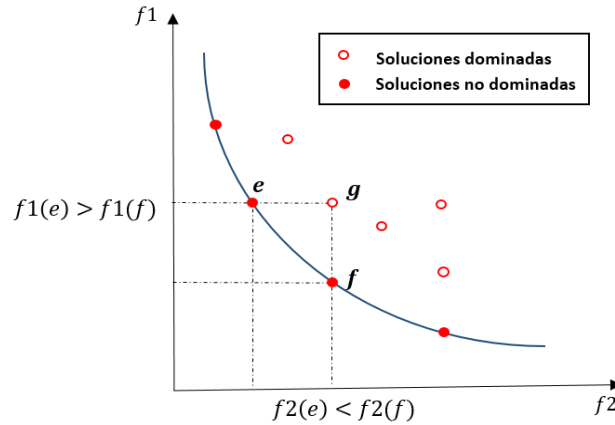


Figura 2 Frontera de Pareto de un problema de Minimización (Adaptado de López [29])

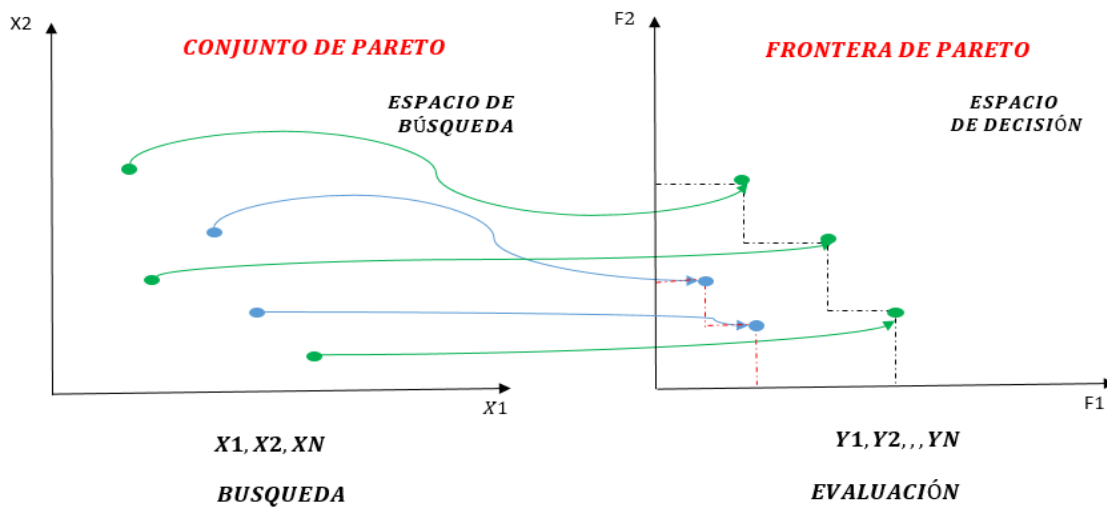


Figura 3 Espacio de búsqueda y espacio objetivo (adaptado de López [29])

2.1.3 FRENTE DE PARETO

El conjunto de todas las soluciones que no son dominadas se conoce como el conjunto óptimo de Pareto o Conjunto de Pareto, se denota como $X^* \subset \mathcal{X}$ y es la solución óptima de un problema multiobjetivo. La evaluación de las soluciones del Conjunto de Pareto forma el Frente de Pareto y se denota como $F(X^*) \subset \mathcal{Y}$. La **Figura 4** muestra ejemplos de las principales formas que pueden presentar los Frentes de Pareto. Teniendo en cuenta un problema donde se busca minimizar los objetivos, los frentes con forma convexa se curvan hacia el interior hacia donde están las mejores soluciones. Por el contrario, los frentes cóncavos se curvan hacia el exterior alejándose de las mejores soluciones. Los frentes lineales no presentan ninguna curvatura y en general corresponden a problemas que son más sencillos de resolver. También se pueden encontrar frentes mixtos, con partes cóncavas,

convexas y líneas, e inclusive se pueden encontrar frentes que son discontinuos en ciertas partes.

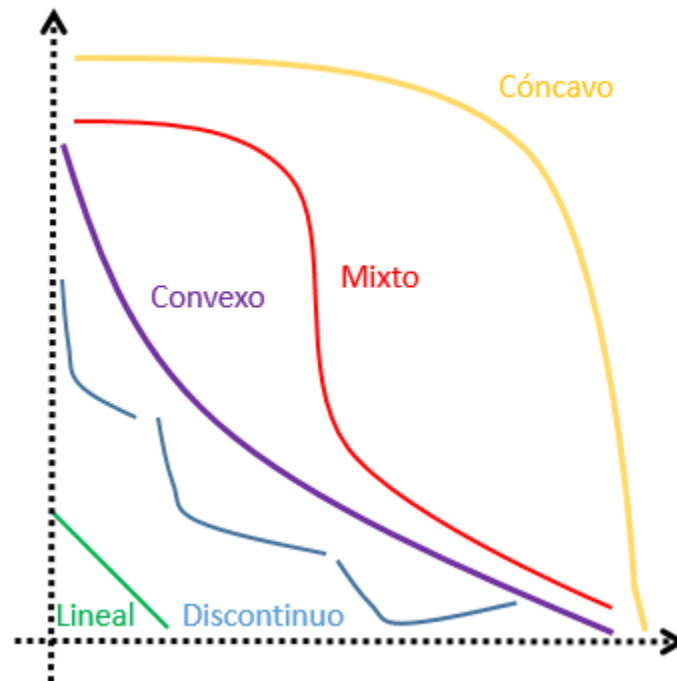


Figura 4 Tipos de frente de Pareto

Como se ha mencionado previamente, el conjunto de Pareto normalmente está compuesto por múltiples soluciones que inclusive puede ser un número infinito de estas. Estos elementos que conforman el Frente o la frontera de Pareto son la solución al problema multiobjetivo. Los algoritmos multiobjetivo buscan obtener un muestreo de soluciones lo más cercanas, es decir, **converjan** al Pareto Optimo o ideal del problema que se está resolviendo, que además sea **representativo** (incluya los extremos) y **diverso** (uniformemente distribuidas por todo el Frente de Pareto) para apoyar al usuario en la toma de decisiones. Los algoritmos de optimización normalmente incluyen medidas para fomentar la diversidad de las soluciones encontradas, generalmente basadas en la distancia de sus objetivos en el diagrama de los múltiples objetivos, más que en la representación misma de la solución (genotipo o fenotipo), para entregar un muestreo más uniforme y completa del Frente de Pareto haciéndolo de la forma más sencilla posible.

2.1.4 PROBLEMAS Y MÉTRICAS DE EVALUACIÓN

En MOPs y MaOPs no se registra aun un criterio único que permita establecer con facilidad si un conjunto de soluciones de aproximación al PF es mejor que otro, en contraste, se han desarrollado varias medidas de calidad que en algunos casos dan resultados contradictorios en su uso. Estas propuestas buscan mediar la cercanía (exactitud o convergencia) de las soluciones encontradas al Pareto Optimo (PO), el

número de soluciones obtenidas (Cardinalidad) y la diversidad de estas. La **Tabla 1** muestra las métricas más popularmente usadas para la evaluación de MOEAs.

En este trabajo se utilizó la distancia generacional invertida (DGI), una métrica que contempla diversidad y convergencia, la cual permitió calcular en un tiempo razonable varios de los experimentos con 80, 90 y 100 objetivos, ya que evaluar por ejemplo el hypervolumen con este número de objetivos se hace computacionalmente prohibitivo. Además, se revisaron otras medidas, pero no se incluyeron en el proceso de evaluación y comparación ya que no son muy utilizadas y reconocidas por la comunidad de optimización de muchos objetivos.

Por otro lado, los problemas o funciones de prueba son diseñados con base en diferentes características (Frentes de Pareto lineales, cóncavos, convexos, degenerados o desconectados, problemas con diferente número de variables, entre otros), buscando presentar diferentes grados de dificultad a los algoritmos que buscan converger a un frente de Pareto óptimo. Dentro de esta investigación se utilizaron 16 funciones de prueba diferenciados en dos familias, la primera de ellas denominada DTLZ (Deb-Thiele-Laumanns-Zitzler) compuesta por siete funciones multimodales (DTLZ1-DTLZ7) [30] y la segunda, la familia WFG (Walking Fish Group) compuesta por nueve funciones escalables (WFG1-WFG9) [31], las cuales ayudaron a determinar la calidad de los algoritmos comparados.

Tabla 1 Métricas de Comparación

Métricas	Aspectos que contempla
Hypervolumen (HV) [32]	- Convergencia - Diversidad
Distancia Generacional Invertida (DGI) [32]	- Convergencia - Diversidad
Distancia Generacional (DG) [32]	- Convergencia
Familia Épsilon (ϵ) [33]	- Convergencia - Diversidad - Cardinalidad
Indicador Delta (Δ) [32]	- Diversidad
Spacing (SP) [32]	- Separabilidad

2.1.5 MOEA FRAMEWORK

MOEA es un marco de trabajo desarrollado en Java de código abierto para computación evolutiva que se especializa en la optimización multi objetivo. Incorpora una variedad de algoritmos evolutivos multi objetivo y es extensible para diseñar, desarrollar, ejecutar y probar de forma rápida algoritmos evolutivos multiobjetivo (MOEAs). Cuenta con 25 MOEAs de última generación y más de 80 problemas de pruebas. A la fecha, ha sido utilizado para realizar numerosos

estudios comparativos donde se evalúa la eficiencia, confiabilidad y capacidad de los MOEAs de última generación.

2.2 ESTADO DEL ARTE

Los MaOEAAs surgen ante las dificultades presentes en los MOEAs tradicionales para tratar MaOPs. Por lo anterior, diferentes investigaciones en el mundo de la computación evolutiva se centran en proponer diferentes enfoques para poder solventar las limitaciones de estos algoritmos. Entre los enfoques propuestos se destacan los basados en dominancia, basados en diversidad, basados en indicadores, basados en preferencias, basados en reducción de la dimensionalidad y los basados en descomposición/agregación.

Este último enfoque (descomposición/agregación) genera el interés particular de los investigadores del área (medido en la cantidad de publicaciones de alto impacto a nivel internacional), en donde el algoritmo MOEA/D [9] es el representante central de este enfoque, el cual ha sido sujeto de diversas mejoras [34].

Dentro de esta investigación se propuso un nuevo método para la generación de vectores de peso dentro del algoritmo MOEA/D y así mejorar los inconvenientes presentes en el método original donde el aumento en el tamaño de la población y sus vectores de peso no aproxima adecuadamente la interacción de los objetivos en un espacio amplio de objetivos [34].

A continuación, se hace un reporte de las modificaciones más importantes y recientes realizadas a MOEA/D en relación con los vectores de pesos.

2.2.1 INICIALIZACIÓN DE VECTORES DE PESOS EN MOEA/D

MOEA/D es un algoritmo que descompone un problema de optimización multiobjetivo en varios subproblemas de optimización mono-objetivo. MOEA/D utiliza un método basado en población para optimizar estos subproblemas simultáneamente e ir encontrando el Frente de Pareto de solución al problema. Teniendo en cuenta que la definición de vectores de pesos en MOEA/D tiene un impacto significativo en los resultados del algoritmo, a continuación, se presentan los trabajos previos que han buscado mejorar dicha definición.

En 2014 [35] se propuso el ajuste adaptativo del vector de pesos en una nueva propuesta denominada MOEA/D-AWA. Aquí, los pesos se ajustan periódicamente para que los subproblemas se puedan redistribuir de forma adaptativa y así obtener una mejor uniformidad en las soluciones, además, introduce una población élite externa para ayudar a agregar nuevos subproblemas en regiones dispersas, es decir regiones discontinuas del PF. Esta propuesta se validó en 10 problemas de referencia ZDT y DTLZ en términos de Distancia Generacional Invertida con algoritmos del estado del arte como MOEA/D, adaptive-MOEA/D, paλ-MOEA/D y

NSGA-II. Los resultados experimentales demostraron que MOEA/D-AWA se desempeña significativamente mejor que los algoritmos comparados en términos de diversidad y convergencia.

En 2018 [36], se presentó un enfoque de inicialización con dos tipos de ajustes de vectores de peso para optimización de muchos objetivos denominado MaOEA/D-2ADV. Esta propuesta, después de realizar la primera iteración de evolución, realiza una búsqueda de los vectores de pesos con mejores soluciones y cercanos al PF óptimo; si encuentra un vector que no satisface con ciertas condiciones de calidad, este se elimina y crea nuevos vectores de peso para su remplazo. Después utiliza el mecanismo basado en dominio de Pareto para detectar la efectividad de cada vector, en caso de que un vector este en una dirección no apropiada, se ajusta para adaptarse de mejor forma al PF. Este algoritmo se comparó con MOEA/D-AWA y RVEA usando DGI en los problemas DTLZ hasta de 10 objetivos concluyendo que MaOEA/D-2ADV es efectivo para trabajar en problemas con PF desconectados con grandes cantidades de objetivos (de 4 a 10).

En este mismo año 2018 [37], se propuso la inicialización de vectores de peso usando Self Organizing Maps (SOM) en una propuesta denominada como MOEA/D-SOM. Los vectores de peso normalizados se envían a SOM para crear vecindarios o agrupaciones de estos, luego se escogen los más cercanos al PF basado en distancia euclidiana. MOEA/D-SOM se evaluó en problemas de muchos objetivos usando 16 problemas diferentes con y sin restricciones entre ellos, DTLZ, TOY y MAOP. Sus resultados fueron comparados con los de MOEA/D-AWA, MOEA/D, M2M, entre otros [38] usando DGI. Se observó que la propuesta es superior resolviendo MaOPs con PF degenerados en comparación con los otros algoritmos.

También en 2018 [39], teniendo en cuenta el papel fundamental de los vectores de peso, los cuales permiten asegurar una buena diversidad y convergencia de las soluciones en los diferentes problemas, más aún en problemas con PF complejos (discontinuos o con picos agudos), se identificó que la distribución uniforme de dichos vectores de pesos en MOEA/D no permiten obtener un conjunto de soluciones con buena diversidad. Por lo anterior, los autores proponen el algoritmo multiobjetivo mejorado basado en descomposición con ajuste de pesos adaptativo, IMOEADA. Esta propuesta, primero utiliza el método basado en diseño uniforme y distancia de apiñamiento (Crowding) para generar un conjunto de vectores de peso distribuidos uniformemente, después, de acuerdo con las distancias de las soluciones dominadas adapta los vectores de peso para redistribuirlos en los espacios subobjetivos. El algoritmo además utiliza una estrategia de selección para ayudar a cada espacio subobjetivo a tener como mínimo una solución. Esta propuesta se comparó con algoritmos del estado del arte como NSGA-II, MOEA/D, MOEA/D-AWA, EMOSA, RVEA y KnEA sobre diferentes funciones de prueba

(DTLZ, WFG, UF y ZDT) usando tres métricas de rendimiento, DGI, Hypervolumen (HV) y distancia generacional (DG). Para analizar los resultados utilizaron la prueba no paramétrica de Wilcoxon y con un 95% de significancia se determinó que la propuesta es capaz de encontrar un conjunto de soluciones con mayor diversidad y convergencia que los otros algoritmos comparados.

Teniendo en cuenta que el enfoque PBI (penalty-based boundary intersection) de definición de vectores de peso, obtiene mejores resultados en problemas cóncavos y convexos que la definición aleatoria uniforme de los pesos y el método de Tchebycheff, pero que su rendimiento se degrada en problemas con PF complejos debido a que define valores de penalidad fijos, en 2019 [40] se propone un esquema de penalización adaptativo (AAP) para ajustar dinámicamente el valor de penalización de cada vector de peso durante el proceso evolutivo del algoritmo. Esta propuesta denominada MOEA/D-AAP se evaluó usando seis problemas de referencia (F1 a F6) y se comparó con MOEA/D-DE, MOEA/D-STM donde el enfoque propuesta mejora significativamente los resultados medidos en DGI.

De igual manera, en 2019 [41] se propone MOEA/HD, un método que utiliza una estrategia de descomposición jerárquica. Los subproblemas escalares se ubican en diferentes jerarquías de pesos y la dirección de búsqueda de las soluciones en los subproblemas de menor jerarquía se ajustan adaptativamente basado en los resultados de la jerarquía superior. Esta propuesta se evaluó y comparó con cuatro propuestas del estado del arte MOEA/D-AWA, NSGA-III, MOEA/D-DRA y NSGA-II en los problemas DTLZ, WFG y JY usando DGI e HV y obtiene los mejores resultados en todos los casos evaluados.

2.2.2 ALGORITMOS DE MUCHOS OBJETIVOS SOBRESALIENTES

En los últimos años se han propuesto diversos algoritmos evolutivos para optimización de muchos objetivos (MaOEA) que buscan superar las deficiencias de los tradicionales MOEA [42]. A continuación, se muestran algunos trabajos previos de los enfoques basados en indicadores y en dominio, por su relevancia con los objetivos del proyecto.

2.2.2.1 Algoritmos de muchos objetivos basados en indicadores

En 2016 [43] se presentó SRA, un algoritmo basado en múltiples indicadores que usa ranking estocástico para equilibrar los sesgos de búsqueda de diferentes indicadores. Los experimentos se realizan sobre un gran número de problemas (39 en total) donde se observa que la propuesta tiene un buen rendimiento en términos de DGI e HV en comparación con los algoritmos del estado del arte. Los estudios empíricos también revelan que, en el caso de que un problema requiera que el algoritmo tenga una fuerte capacidad de convergencia, el rendimiento de SRA

puede mejorarse aún más mediante la incorporación de un archivo basado en dirección para almacenar soluciones convergentes y mantener la diversidad.

También en 2016 [18] se presentó ϵ -MOABC, un método de optimización de colonias de abejas artificiales multiobjetivo basado en indicadores de rendimiento. ϵ -MOABC gestiona un archivo externo sobre la base del dominio de Pareto y los indicadores de preferencia para guardar las soluciones no dominadas producidas en cada generación. El algoritmo propuesto demuestra ser competitivo en el tratamiento de problemas de optimización de muchos objetivos en comparación con otros algoritmos de última generación para las instancias de prueba CEC09, LZ09 y DTLZ.

2.2.2.2 Algoritmos de muchos objetivos basados en dominio

En 2015 [44] se propuso KnEA, un MOEA que usa los puntos de inflexión para dar preferencia a las soluciones no dominadas, si no se dan preferencias explícitas de parte del usuario. El algoritmo identifica un punto de inflexión dentro del vecindario de cada solución en el frente no dominado, sin introducir mecanismos adicionales de mantenimiento de diversidad reduciendo considerablemente la complejidad computacional en comparación con otros MOEAs existentes para optimización de muchos objetivos. Los resultados experimentales en 16 problemas de prueba demuestran la competitividad del algoritmo propuesto en términos de calidad de la solución y eficiencia computacional.

2.2.3 NSGA-III

La versión tres del Algoritmo Genético de Ordenamiento No Dominado es una extensión mejorada de NSGA-II para problemas de muchos objetivos (4 o más) [45]. El algoritmo mantiene el esquema general de NSGA-II, pero le incorpora una técnica de puntos de referencia [22]. Esta técnica sustituye al segundo criterio de selección basado en distancia de apiñamiento (crowding) entre soluciones no dominadas. Estos puntos de referencia representan las regiones del PF que deben ser descubiertas, con el objetivo de mantener la diversidad en la población resultante de cada generación. Para discriminar entre las soluciones no dominadas se utiliza una función de utilidad, cuyo valor indica la relevancia de una solución para aproximar un punto de referencia. En la **Figura 5**, se muestra un ejemplo de la diferencia entre el funcionamiento de la distancia de apiñamiento que emplea NSGA-II y la técnica empleada por NSGA-III. Todos los círculos representan los individuos del primer frente no dominado de la población, y los círculos negros representan los individuos seleccionados para la siguiente generación. La efectividad de este algoritmo fue analizada en varios problemas de hasta 15 objetivos. Los resultados experimentales mostraron que posee un desempeño

competitivo respecto a dos versiones de MOEA/D (MOEA/D TCH y MOEA/D-PBI) en problemas de optimización continuos.

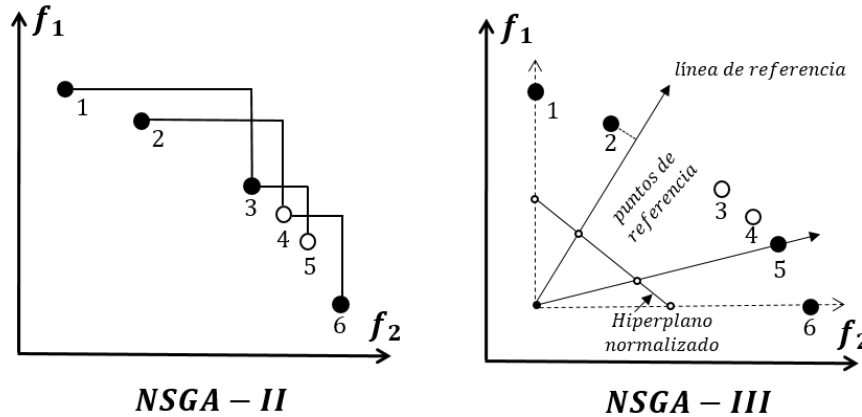


Figura 5 Ejemplo de la diferencia entre el funcionamiento de NSGA-II y NSGA-III

El **Algoritmo 1** presenta los pasos de NSGA-III en cada una de sus generaciones para poder generar una nueva población (P_{t+1}) con base en la población actual P_t . En el paso 2 se crea una población Q de N hijos con base en la población P (P_t). Luego en el paso 3 se unen la población de padres P con la de hijos Q en R . Después, en el paso 4 se ordenan por dominio de Pareto las soluciones en R , empezando por el F_1 que son soluciones no dominadas, luego el frente F_2 con soluciones que son dominadas por al menos una solución de F_1 y así sucesivamente. En los pasos 5 a 9 se inicializa S como un conjunto vacío de soluciones y luego, uno a uno los Frentes se van agregando a S hasta que se iguale o supere el número N de soluciones que debe tener la nueva población (P_{t+1}). En este punto S puede tener N soluciones o un número mayor. Si S tiene N soluciones (pregunta del paso 11) la población ($P_t = S$) y el procedimiento retorna esta población S con un conjunto de soluciones élite. Si por el contrario S tiene más de N soluciones (sino del paso 13), primero se registra el ultimo frente que se adicionó a S , es decir el último valor i del proceso iterativo previo (ver paso 10) y se realizan los pasos 14 a 20.

En el paso 14 se arma P con las soluciones de todos los frentes excepto el último que se agregó a S (quedando en P las soluciones élite de la población anterior y los hijos generados en la generación actual). Con esto se asegura que $|P|$ siempre sea $\leq N$ y que faltan K (ver paso 15) soluciones por incluir en P para tener la nueva población (P_{t+1}). Los pasos siguientes 16 a 20 buscan definir las K soluciones (las que ayuden a mejorar la diversidad de la población) del último frente adicionado a S (F_{ultimo}) que se deben agregar a P para completar la población.

En NSGA-II, para completar las K soluciones y promover diversidad se usa la distancia de Crowding, pero este operador es muy costoso computacionalmente con muchos objetivos y no aporta la diversidad necesaria, por lo que se reemplaza por un proceso de fomento a la diversidad en nichos (grupos) de soluciones alrededor de puntos de referencia.

Algoritmo 1 Procedimiento de la generación en NSGA-III	
Entrada:	<ul style="list-style-type: none"> • N: Tamaña de la población, • O: Lista de M objetivos a optimizar, // se asume minimización de todos ellos • P: Población en iteración i
Salida: Población en la iteración $i+1$ para los M objetivos	
<ol style="list-style-type: none"> 1. Inicio 2. Cree una población Q de N hijos usando como operador de selección de padres el torneo binario, cruce de un punto y mutación a nivel de bits para variables binarias, cruce binario simulado (SBX) con un índice de distribución grande (para crear hijos cercanos a los padres) y mutación polinomial para variables reales, y evalúe los O objetivos de todos los hijos 3. $R = P \cup Q$ // Agregue la población P y Q en R 4. $(F_1, F_2, \dots) =$ Ordenar R basado en dominio de pareto 5. $S = \emptyset, i = 1$ // Se agregan tantos frentes (F_i) en S hasta que se iguale o supere el numero maximo (N) de soluciones para la nueva población 6. Haga 7. $S = S \cup F_i$ 8. $i = i + 1$ 9. Mientras que $S_t \geq N$ 10. Ultimo = i Fultimo es el frente para ser incluido S 11. Si $S \geq N$ Entonces 12. Retorne S 13. Sino 14. $P = \cup_{j=1}^{ultimo-1} F_j$ // se unen los P frentes que no superan las N soluciones 15. $K = N - P$ // Número de soluciones que falta incluir en P desde F_{ultimo} 16. Crear conjunto de referencia Z^r: 17. Normalizar(S_t, Z^r, Z^s, Z^a) 18. $[\pi(s), d(s)] =$ Asociar (S_t, Z^r) // Asociar a cada <i>miembro</i> s de S con un punto de referencia mas cercano z en Z: $\pi(s)$ y guarde ademas, d: Distancia entre s y $\pi(s)$ 19. $V =$ Contar (S, F_{ultimo}, Z^r) // Calcule el número s de soluciones de S que están asociados a cada punto de referencia z en z y que no están en F_{ultimo}, es decir, sólo los que están en P_{t+1} 20. $P =$ Armar nichos ($K, V, \pi, d, Z^r, F_{ultimo}, P$) // Uno a uno incluya desde F_{ultimo} en P las K soluciones que hacen falta para completar la población 21. Fin si 22. Retornar P 23. Fin 	

Algoritmo 1 NSGA-III

En el paso 16 se crea el conjunto de referencia z^r , paso que podría ser el primero del algoritmo ya que no depende de ningún otro paso de este. Estos puntos de

referencia pueden definirse de forma estructurada o ser suministrados por parte del usuario con base en sus preferencias (en este caso deben estar normalizados o se deben normalizar en este paso). Para definir estos puntos de referencia de la forma estructurada, en NSGA-III se usa la propuesta por Das y Dennis [22] en que usa un enfoque sistemático para ubicar puntos en un hiperplano normalizado que está igualmente inclinado en todos los ejes de los objetivos y se intercepta en uno en cada uno de estos ejes.

La **Figura 6** muestra la distribución de 15 puntos en 3 dimensiones de objetivos. La cantidad de puntos de referencia H que se crean en el hiperplano se calcula con base en la combinatoria presentada en la **Ecuación 2**. Donde H es la cantidad de puntos de referencia definidos en el hiperplano, M es la cantidad de funciones objetivo y p es la cantidad de divisiones en las abscisas.

$$H = \binom{M+p-1}{p} = \frac{(M+p-1)!}{(M-1)! \times p!}$$

Ecuación 2 Puntos de referencia Hiperplano

Para el caso de la **Figura 6** se tienen 3 objetivos ($M = 3$) y el valor de p se define en 4, por lo que la combinatoria de 6 en 4 es igual a 15 ($H = 15$). Como se observa en la figura, estos puntos están distribuidos sobre el hiperplano, buscando con ello que se cubra extensamente el Frente de Pareto óptimo, fomentando diversidad. En la figura también se muestra una “Línea de referencia” que se representa como el vector que va del origen y pasa por un punto de referencia, esta línea se usa luego para calcular la distancia de las soluciones hacia los puntos de referencia. Una recomendación importante de NSGAIII es que el valor de N (tamaño de la población) sea aproximadamente igual o igual a H ($N \approx H$).

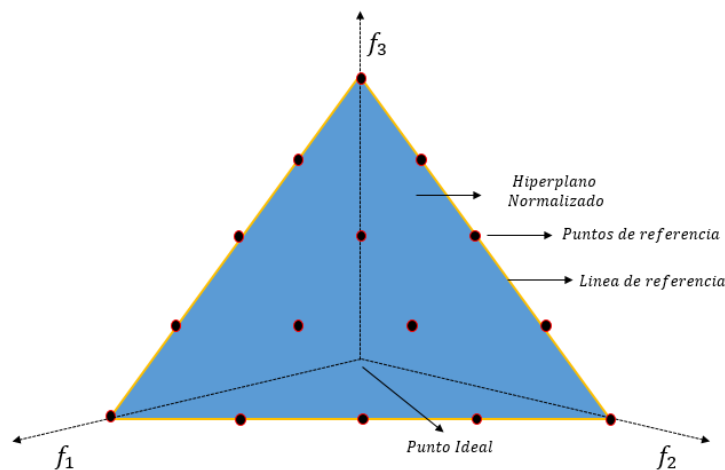


Figura 6 Distribución de 15 puntos de referencia para un problema con 3 objetivos tomado de [46].

En el paso 17 se realiza la normalización de los miembros de la población S . Hay que tener en cuenta que, en este punto, S tiene los elementos que ya se incluyeron en P y los puntos del último Frente (F_{ultimo}) de los que se debe escoger los que van a ingresar a P . Lo primero que se hace es calcular el punto ideal (z^{min}) que corresponde al valor mínimo para cada uno de los objetivos $z^{min} = (z_1^{min}, z_2^{min}, \dots, z_M^{min})$, y el valor máximo en cada uno de los objetivos $z^{max} = (z_1^{max}, z_2^{max}, \dots, z_M^{max})$. Luego los valores de los objetivos de cada solución se normalizan usando la técnica *Min – Max*, donde cada valor normalizado x' de x es igual a $(x - min) / (max - min)$. Como este proceso de normalización se realiza en cada generación usando los puntos extremos (*mínimo y máximo*), NSGA-III adaptativamente mantiene soluciones diversas en S en cada generación y le permite resolver problemas con objetivos que tengan diferente escala.

En el paso 18 se realiza el proceso de asociación. Ese proceso busca definir para cada solución de S que ya tiene sus objetivos normalizados el punto de referencia z de Z^r al que está más cerca ($\pi(s)$). Para determinar dicho punto, se calcula la distancia de la línea de referencia de cada punto en Z^r perpendicular a la solución conforme se muestra en el lado izquierdo de la **Figura 7**.

En esta figura se pueden observar 8 soluciones en color naranja y 5 puntos de referencia en el hiperplano normalizado. La solución 1 (s_1) tiene menor distancia, $d(s)$, hacia la línea de referencia del punto z_1 . Las soluciones s_4, s_5 y s_6 tienen su menor distancia a la línea de referencia del punto z_3 y la elipse que envuelve estas tres soluciones marca el nicho que este punto de referencia tiene. Ese mismo proceso de asociación se hace con problemas que tienen más dimensiones, la imagen del lado derecho muestra las tres líneas de referencia (vectores en color negro) más cercanas a la solución x (en rojo). Allí se procede a calcular las perpendiculares al punto de solución y aquel que sea menor se convierte en el valor $d(x)$ y el punto de referencia el $\pi(x)$. Estos cálculos se realizan usando la **Ecuación 3**, donde s es una solución de S y w es la línea de referencia a los puntos z y Z^r .

$$d(s, w) = \|(s - w^T s w / \|w\|^2)\|$$

Ecuación 3 Puntos de referencia NSGA-III

Después de asignar las distancias y la asociación de cada solución s a los puntos de referencia, pueden quedar puntos de referencia sin soluciones, con una solución o con más de una solución. En el paso 19 se calcula el número s de soluciones de S que están asociados a cada punto de referencia z en Z^r y que no están en F_{ultimo} , es decir, sólo las soluciones que están en P (P_{t+1}). Este resultado se almacena en un vector de vecinos o nicho de cada punto de referencia que se denomina V .

El paso 20 se encarga de incluir una a una las K soluciones que están en F_{ultimo} y aportan más diversidad a las soluciones que ya están en P (P_{t+1}).

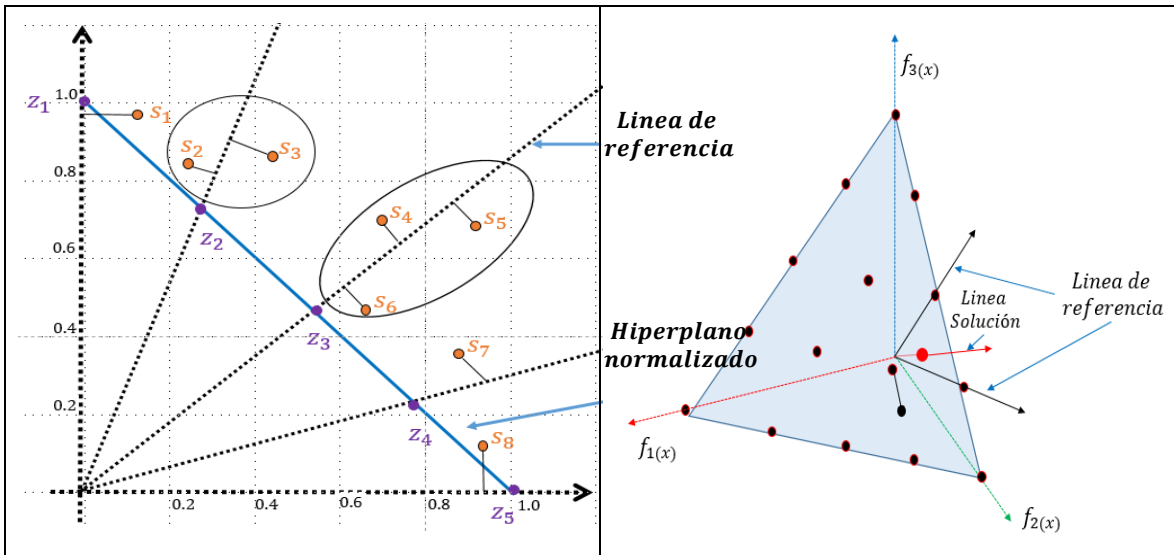


Figura 7 Calculo de distancia para asociar soluciones a los p de referencia 2D,3D
Tomado de [46]

La efectividad de este algoritmo fue analizada en varios problemas hasta con 15 objetivos. Los resultados experimentales mostraron que posee un desempeño competitivo respecto a dos versiones de MOEA/D (una con el modelo de escalarización de Tchebycheff y la otra con el método de la intersección de límites basada en penalidades, PBI) en problemas de optimización continuos.

2.2.4 MOEA/D-DE

En 2009, Li y Zhang [12] propusieron una versión de MOEA/D basada en la evolución diferencial (Algoritmo Evolutivo Multiobjetivo Basado en Descomposición con Evolución Diferencial, MOEA/D-DE). Esta propuesta se distingue en varios aspectos de MOEA/D. En primer lugar, utiliza la mutación polinomial y los operadores de la evolución diferencial que le permiten mejorar el desempeño en la búsqueda de las mejores soluciones en el PF. Y, en segundo lugar, agrega dos parámetros al algoritmo para evitar la pérdida de diversidad.

Una de las características negativas de MOEA/D, es que una solución puede reemplazar a todos sus vecinos. Esto provoca que se pierda la diversidad en la región que la contiene y resulte difícil para los operadores de variación que generen luego una mejor solución. Para evitar esta dificultad, MOEA/D-DE utiliza el parámetro, η_r el cual establece el número máximo de soluciones que pueden ser reemplazadas por un mismo individuo.

La **Figura 8** presenta un ejemplo del problema de pérdida de diversidad en MOEA/D, partiendo de una población de seis soluciones que en su proceso de actualización se asocia cada una de ellas a un vector de pesos $w_i, i = 1 \dots 6$, la solución x_7 posee mayor valor en función de utilidad minimizando que las soluciones de la vecindad ($x_i, i = 1, \dots, 6$). Debido a esto x_7 pasa a ser la solución de 6 subproblemas y se repite igual número de veces en la población, disminuyendo así la diversidad de la misma.

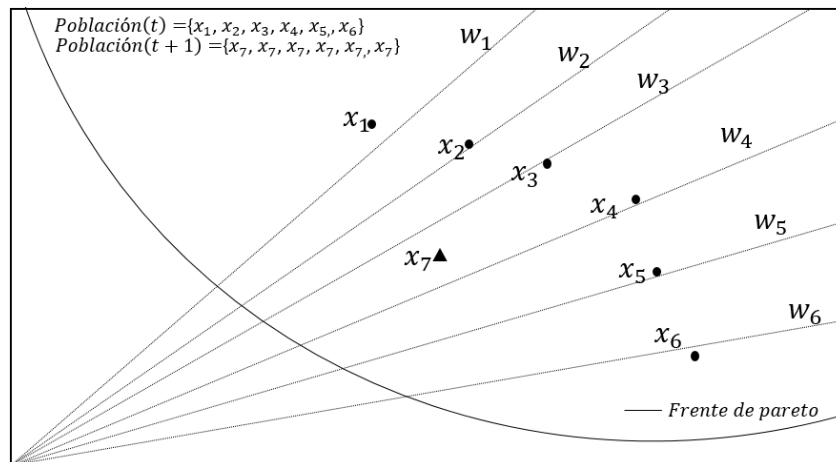


Figura 8 Problema de pérdida de diversidad en MOEA/D

MOEA/D-DE por su parte, permite, con cierta probabilidad, que un nuevo individuo sea generado a partir de soluciones de distintas vecindades. Además, limita a un valor predefinido el número de soluciones que pueden ser reemplazadas por un mismo individuo. Sin embargo, MOEA/D-DE busca el óptimo de cada subproblema de manera independiente, suponiendo que esto genera el mejor conjunto de soluciones de manera global, en asociación a un vector de pesos, tal y como se muestra en la **Figura 9**. MOEA/D-DE genera una nueva solución la cual busca en que subproblema este individuo puede sustituir a su solución actual, en el ejemplo de denota que x_5 es la nueva solución que se asocia al subproblema w_3 actualizando así a la solución x_3 .

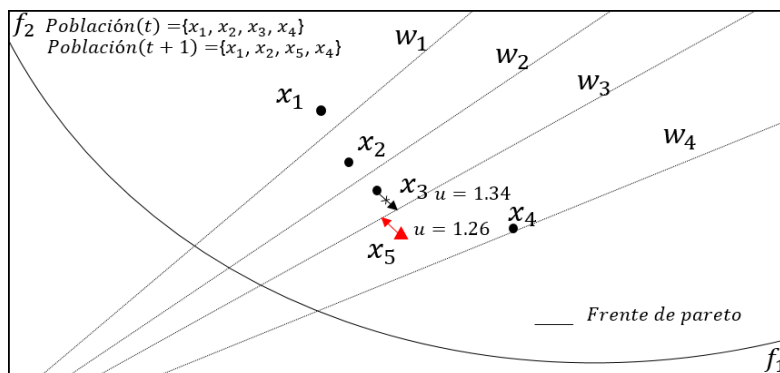


Figura 9 Actualización de soluciones MOEA/D-DE

A continuación se describe cada una de las etapas de MOEA/D-DE **Algoritmo 2**

<p>Algoritmo 2 MOEA/D-DE</p>
<p>Entrada:</p> <ul style="list-style-type: none"> • Problema: Minimizar $F(x) = \text{Minimiza } (f_1(x), f_2(x), \dots, f_m(x))^T$, sujeto a $x \in \Omega$ donde $F: \Omega \rightarrow \mathbb{R}^m$, consiste en m funciones objetivo de valor real presentes en el espacio objetivo, Ω es el espacio de decisión (variable), $\Omega = \{x \in \mathbb{R}^n h_j(x) \leq 0, j = 1, \dots, m\}$ y h_j son funciones objetivo continuas • T: número de vectores de peso en el vecindario de cada vector de peso • δ: probabilidad de que las soluciones de los padres se seleccionen del vecindario • η_r: número máximo de soluciones remplazadas en cada generación. • A Criterio de parada
<p>Salida:</p> <ul style="list-style-type: none"> • Aproximación al conjunto de Pareto (PS) : $\{x^1, \dots, x^N\}$ • Aproximación al PF: $\{F(x^1), \dots, F(x^N)\}$
<p>Inicialización</p> <ol style="list-style-type: none"> 1. Calcule las distancias euclidianas entre dos vectores de peso cualquiera y luego calcule los vectores de peso más cercanos a cada vector de peso. Para cada $i = 1, \dots, N$ establezca $B(i) = \{i_1, \dots, i_T\}$ donde w^{1T}, \dots, w^{iT} son los T vectores de peso mas cercanos a w^i. 2. Genere una población inicial x^1, \dots, x^N aleatoria en Ω y 3. Para cada $i = 1, \dots, N$ calcule $F(x^i)$. 4. Inicialice $z = (z_1, \dots, z_m)$ donde $z_j = \min_{1 \leq i \leq N} f_j(x^i)$ <p>5. Actualización</p> <ol style="list-style-type: none"> 6. Para $i = 1, \dots, N$ realice Selección de rango de acoplamiento / actualización: Genere uniformemente un número aleatorio desde $[0,1]$. Luego establezca: $P = \begin{cases} B(i) & \text{if aleatorio} < \delta \\ \{1, \dots, N\} & \text{de lo contrario} \end{cases}$ 7. Reproducción: Conjunto $r_1 = i$ y seleccione aleatoriamente 2 índices r_2 y r_3 en P, genere una solución \bar{y} desde x^{r_1}, x^{r_2} y x^{r_3} por medio del operador DE, luego realice una mutación en \bar{y} con probabilidad p_m para producir una nueva solución y. 8. Reparación: Si un elemento de y esta fuera del limite, Ω, su valor se restablece para que sea un valor seleccionado al azar dentro del limite. Calcular $F(y)$. 9. Actualización de z: Para cada $j = 1, \dots, m$, if $f_j(y) < z_j$, establezca un conjunto $z_j = f_j(y)$. 10. Actualización de soluciones: establezca $c = 0$ y luego haga lo siguiente: 11. Mientras que P no está vacío hacer: <ol style="list-style-type: none"> a. Si $c = \eta_r$ vaya al Paso 3. b. Elegir aleatoriamente e indexar j de P. Si $g(y w^j, z) \leq g(x^j w^j, z)$ entonces <ol style="list-style-type: none"> a. establezca $x^j = y, F(x^j) = F(y)$, and $c = c + 1$. b. retire j de P. 12. Terminar Mientras 13. Criterio de parada: Si se cumple el criterio de parada, entonces detenga y envíe $\{x^1, \dots, x^N\}$ and $\{F(x^1), \dots, F(x^N)\}$. De lo contrario, dirigirse a Paso 2

Algoritmo 2 MOEA/D-DE

MOEA/D-DE recibe como parámetros de entrada T que es igual al numero de vectores de peso para resolver el problema, además de δ : probabilidad de que las soluciones de los padres se seleccionen del vecindario y η_r : número máximo de soluciones remplazadas en cada generación con ello obtiene como salida la

aproximación al conjunto de Pareto PS y PF , después de ello inicia con el paso 1 donde calcula las distancias euclidianas entre dos vectores de peso cualquiera y luego calcula los vectores de peso más cercanos a cada vector de peso. Donde w^{1T}, \dots, w^{iT} son los T vectores de peso más cercanos a w^i .

Después en el paso 3 genera una población de soluciones aleatorias dentro del espacio de decisión y para cada uno de ellos calcula su aptitud, en el paso 4 inicializa los puntos de referencia $z = (z_1, \dots, z_m)$. En el paso 5 denominado actualización, para cada una de las soluciones realiza los procesos de selección y métodos de reproducción basado en el método de evolución diferencial, después de ello si existen soluciones no aptas se realiza el paso 8 denominado reparación se reparan las soluciones basado en las restricciones del problema, luego en el paso 10 de actualización se actualizan los puntos de referencia basado en los datos de la población nueva, después mientras la Población no esté vacía entonces indexa cada una de las soluciones aptas aleatoriamente si la solución es mejor y domina a la solución anterior en el frente de Pareto sale del procedimiento de lo contrario regresa al paso 5 [12].

2.2.5 ARREGLOS DE COBERTURA

Los arreglos de cobertura (Covering Arrays, CA) han sido tradicionalmente utilizados en el diseño de experimentos, en campos como la agricultura, la medicina, la biología, el diseño de materiales entre otros, donde se requiere probar combinaciones de distintos factores sin realizar una combinación exhaustiva de los mismos debido a restricciones de costo, tiempo y esfuerzo, y más recientemente en pruebas de software y hardware [47].

Los CA aparecen como una alternativa a las pruebas exhaustivas, ya que dependiendo de un parámetro denominado fuerza (t) del arreglo se puede controlar el número mínimo de interacciones entre los componentes que se desean evaluar y se puede reducir al mínimo posible el número de casos de prueba que se deben realizar. Un CA puede ser expresado a partir de la notación $CA(N: t, k, v)$, que representa una matriz de tamaño $N \times k$, donde N se refiere al número de pruebas, k es el número de parámetros, factores o variables, v es el alfabeto que indica el número posible de valores que puede tomar cada componente. Por ejemplo, si se tiene un arreglo cuyos valores son sólo 0 y 1 se dice que tiene un alfabeto binario. El valor de t es la fuerza o grado de interacción entre los parámetros. La característica especial que tienen los CAs es que cualquier conjunto t de columnas que se extraigan del arreglo, contiene todas las combinaciones posibles de v^t tuplas en al menos una de las filas [26].

En la **Figura 10** se presenta un CA de fuerza dos. Es posible observar en el arreglo, que para cualquier conjunto de 2 columnas ($t=2$), siempre existen las posibles tuplas [(0,0), (0,1), (1,0) y (1,1)].

$$CA(N = 5: t = 2, k = 4, v = 2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Figura 10 Ejemplo CA fuerza 2 (Fuente propia)

La creación de CAs es un problema complejo, para el que se han propuesto métodos exactos, aproximados, greedy y meta-heurísticas, siendo los últimos los que a la fecha presentan mejores resultados [48]. En esta investigación es de interés el uso (no la creación) de CAs para la inicialización de los vectores de peso del algoritmo MOEA/D con el fin de tener en cuenta la posible interacción de los objetivos. Con el número de filas del CAs se permite además establecer el tamaño de la población.

CAPÍTULO 3

3 PROPUESTA

3.1 MOEA/D-DE-ACA

MOEA/D-DE al igual que MOEA/D sigue dependiendo de la apropiada distribución de los vectores de pesos para encontrar mejores soluciones. Esto ha motivado al desarrollo de nuevos métodos que puedan incorporarse en estos algoritmos para generar vectores de pesos mejor distribuidos. Mediante un enfoque de descomposición un MaOPs se descompone en varios problemas de optimización escalares (Single Objective Problems, SOPs) que se definen de la siguiente manera.

$$\min g(x|w, z^*) = \max_{1 \leq i \leq q} w_i | f_i(x) - z_i^* |, \text{ tal que } x \in \Omega$$

Ecuación 4 Definición SOPs

donde $z^* = (z_1^*, z_2^*, \dots, z_q^*)$ constituye un punto referencia, o sea, el valor óptimo de la función i –ésima para una de las soluciones pertenecientes a Ω .

Al igual que MOEA/D y MOEA/D-DE, la propuesta para incorporar de manera fija los vectores de peso denominada **Algoritmo Evolutivo Multiobjetivo Basado en Descomposición con Evolución Diferencial y Arreglos de Cubrimiento Aumentados (MOEA/D-DE-ACA)** archiva la mejor solución encontrada por cada subproblema a optimizar. Para cada subproblema, se genera una nueva solución utilizando operadores evolutivos que involucran la información de la vecindad del subproblema en cuestión. La mejor solución encontrada es actualizada si la nueva solución es mejor. Esta mejor solución encontrada hasta el momento por el subproblema j –ésimo es compartida por los vecinos del subproblema, esto permite actualizar la mejor solución encontrada por estos subproblemas vecinos. Este principio de explotación u optimización local constituye la fortaleza del método.

Formalmente, sea w^1, \dots, w^N un conjunto de vectores de pesos fijos obtenidos de un ACA y z^* un punto de referencia, la función objetivo del j –ésimo subproblema: donde $w^j = (w_1^j, \dots, w_N^j)$. De esta forma, por cada solución no dominada x^* existe un vector de pesos w tal que x^* es una solución óptima y cada solución óptima constituye una solución no dominada del problema definido en la ecuación (3). Por tanto, MOEA/D-DE-ACA optimiza los N SOPs en una sola iteración. Variando w se obtiene diferentes soluciones no dominadas. Estas soluciones se mantienen utilizando un archivo externo. En este sentido, cuando se generan nuevas soluciones se verifica cuál de ellas domina alguna de las existente en el archivo externo y se reemplazan aquellas que resulten dominadas.

Con vectores de pesos bien distribuidos aumenta la probabilidad de obtener soluciones óptimas en los subproblemas que a su vez estén distribuidas sobre el PF.

3.2 COVERING ARRAYS Y AUGMENTED COVERING ARRAYS

Un Covering Array $CA(N; t; k; v)$ es una matriz $N * K$ en los símbolos v de modo que cada subarreglo $N * t$ contiene como fila cada $t - tupla$ sobre los símbolos v al menos una vez. El número de la matriz de cobertura $CAN(N; t; k; v)$ define el valor mínimo de N tal que existe un $CA(N; t; k; v)$.

Los CA se han utilizado de manera exitosa en las pruebas de Software y Hardware [49], En este contexto, las N filas de un CA representan una prueba, cada una de las k columnas indica una variable utilizada en el proceso de prueba, el orden o el alfabeto v indica los valores posibles para cada variable (es decir, cada celda de la matriz contiene un valor en el rango $\{0, \dots, v - 1\}$, y el parámetro t se llama tamaño de interacción e indica que todas las combinaciones de t variables se muestrean al menos una vez.

En la **Figura 11** se presenta un CA de fuerza dos. Es posible observar en el arreglo, que para cualquier conjunto de 2 columnas ($t=2$), siempre existen las posibles tuplas [(0,0), (0,1), (1,0) y (1,1)].

$$CA(N = 5; t = 2, k = 4, v = 2) = \begin{matrix} & & & & k \\ & & & & \left. \begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{array} \right\} \\ & & & & \left. \begin{array}{cccc} \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array} \right] & \left. \vphantom{\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array}} \right\} N \\ & & & & \left. \begin{array}{cc} t & v \end{array} \right\} \end{matrix}$$

Figura 11 Ejemplo CA fuerza 2

De manera general, los CA pueden verse como un mecanismo de muestreo en muchos contextos posibles [50], [51], en este trabajo se utiliza el CA para muestrear soluciones de múltiples ecuaciones lineales diofánticas con coeficientes de unidad (Linear Diophantine Equations with Unit coefficients, $LDEU$). Las soluciones de una $LDEU$ se pueden usar en la optimización de múltiples objetivos para asignar los pesos a los objetivos [52],[37],[23], los pesos tienen valores de 0 a 1 y la suma de todos los pesos debe ser 1. La generación de pesos utilizando una solución $LDEU: x[0] + \dots + [k + 1] = \alpha$ se hace fácilmente dividiendo la solución por α . El valor de α define la granularidad de los pesos, es decir, la tosquedad o suavidad de los

pesos, por ejemplo, si $\alpha = 10$ la granularidad está en décimas y si $\alpha = 100$ la granularidad está en centésimas.

A continuación, se explica el muestreo de las soluciones *LDEU* en el dominio de los enteros y no en el dominio de los vectores de pesos (reales). Cada fila en un *CA* puede verse como una de las posibles soluciones de una *LDEU* correspondiente. La solución *LDEU* asociada a una fila de un *CA* con k columnas y de orden v es: $x_0 + \dots + x_{k-1} = \alpha$ donde el valor de α se obtiene como la suma de los elementos en una fila del *CA*, por lo que sus valores se describen mediante $\alpha \in \{0, \dots, k * (v - 1)\}$. El valor cero corresponde a una fila con todas las celdas de la fila igual a cero, y $k * (v - 1)$ corresponde a una fila con todos los valores de celda iguales a $v - 1$. El número de soluciones posibles para un valor específico de α de la *LDEU* $x_0 + \dots + x_{k-1} = \alpha$ se describe por $\binom{\alpha + k - 1}{k - 1}$ y el acumulado para todos los α se presenta en el **Ecuación 5**.

$$\sum_{\alpha=0}^{k(v-1)} \binom{\alpha + k - 1}{k - 1} = \sum_{i=k-1}^{k v - 1} \binom{i}{k - 1}$$

Ecuación 5 Número de Soluciones CA

El número total de soluciones posibles de todas las *LDEU* crece exponencialmente de acuerdo con los valores de k (el número de columnas en un *CA* que corresponden al número de objetivos) y el valor de v (el alfabeto v al final determina el conjunto de posibles valores α que transitivamente define la granularidad de la asignación de pesos). Por otro lado, el número de soluciones muestreadas por un *CA* es el número de filas que están delimitadas asintóticamente por la expresión que define un *CAN* en [53],[54],[55] que corresponde al límite de Stein-Lovász-Johnson (SLJ). Sean $t, k, AND v$ números enteros con $(k \geq t \geq 2) \wedge (v \geq 2)$ Entonces cuando $k \rightarrow \infty$, el número de filas del *CA* es menor igual que la **Ecuación 6**.

$$CAN(t, k, v) \leq \frac{\log\left(\frac{k}{t}\right) + t \log(v)}{\log\left(\frac{v^t}{v^t - 1}\right)}$$

Ecuación 6 Número de soluciones Muestreadas

En este sentido, el número de soluciones muestreadas por las filas de un *CA* es mucho más pequeño que el número de soluciones posibles que corresponden a todas las soluciones muestreadas de una *LDEU*. Por ejemplo, para $k = 10, v = 5$ y $t = 2$, una *CA* posible tendrá 36 filas. En la **Tabla 2** se puede ver en las primeras 10 columnas el contenido del *CA* y en la 11ª columna el valor de α . Se puede ver que en la **Tabla 2** solo se muestrea un pequeño número de valores α

posibles, el número total de clases de α es $1 + k(v - 1) = 41$ pero solo los valores para $13 \leq \alpha \leq 30$ son muestreados (17 valores α diferentes). Es deseable muestrear un mayor rango de valores α y en lo posible que estos valores sean pequeños para muestrear interacciones entre menos objetivos.

Tabla 2 CA (36;2,10,5)

4	0	0	2	0	4	1	0	0	2	13
1	0	1	3	2	2	0	3	0	2	14
2	1	1	0	0	2	3	1	2	3	15
0	0	4	2	1	0	4	1	1	3	16
3	3	4	0	2	2	1	0	1	0	16
1	4	2	1	0	0	1	4	3	0	16
0	2	3	3	0	3	1	2	1	1	16
2	2	0	3	3	0	0	0	2	4	16
0	2	2	1	2	1	3	0	4	2	17
2	1	0	4	1	1	4	0	3	1	17
0	1	0	4	2	4	2	2	2	0	17
2	2	2	0	1	3	4	3	0	0	17
1	3	0	1	1	3	3	1	2	2	17
1	0	1	0	3	1	4	2	3	2	17
2	1	1	1	4	1	1	3	1	3	18
0	3	3	0	4	1	0	4	0	4	19
4	3	1	0	2	0	2	3	4	1	20
3	2	1	2	1	4	0	4	3	0	20
3	0	2	1	2	3	2	1	3	4	21
1	0	2	2	4	2	3	4	2	1	21
3	4	4	1	3	4	0	1	0	1	21
2	3	2	3	0	4	2	2	0	3	21
1	1	4	2	0	3	0	2	4	4	21
1	2	3	4	1	2	2	0	4	3	22
3	1	3	4	4	0	3	2	0	2	22
2	0	3	4	3	4	1	1	4	0	22
1	4	0	0	3	4	3	3	1	4	23
4	4	0	3	1	2	1	2	2	4	23
2	4	4	2	2	1	2	4	1	2	24
0	3	3	2	3	2	2	3	3	3	24
4	1	2	4	3	3	0	4	1	3	25
4	2	4	3	4	1	3	1	3	0	25
0	4	1	4	4	3	2	0	4	4	26
4	4	3	1	2	2	4	2	2	3	27
3	3	4	4	0	1	4	3	2	4	28
3	1	0	3	4	4	4	4	4	3	30

En la **Figura 12** se muestra como el CA no logra una cobertura total del PF. Los 5 vectores de peso no tienen la suficiente diversidad para muestrear totalmente el PF, hecho que se resalta con las flechas rojas, las cuales muestran espacios donde no se podrían obtener soluciones y que afecta el rendimiento de los algoritmos basados en descomposición en la medida que los problemas se hacen más complejos, por ejemplo, problemas con PF discontinuos o con formas mixtas.

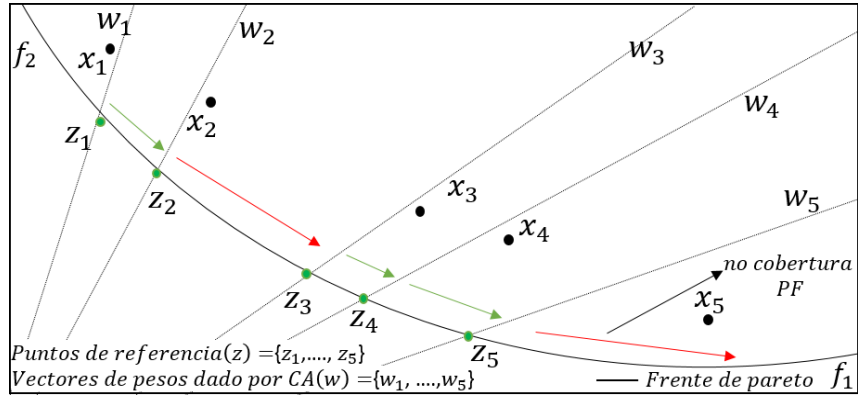


Figura 12 Cobertura PF de un CA

Esta es la razón por la que en esta investigación se decidió utilizar una variante de CA, denominado arreglo de cobertura aumentada (ACA). Un ACA $(N; t, k, v)$ se construye utilizando un CA con un alfabeto inferior a v agregando las filas necesarias que satisfacen la propiedad de cobertura como se observa en la **Figura 13**. Este muestreo con el ACA está más uniformemente distribuido sobre el espacio de objetivos.

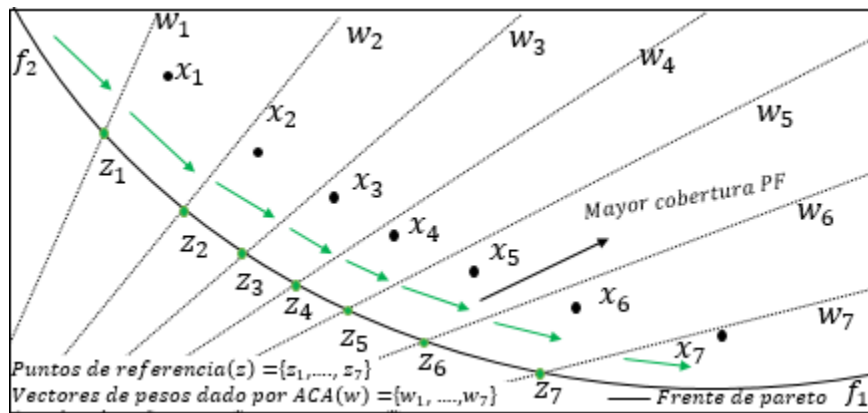


Figura 13 Cobertura PF de un ACA

Por ejemplo, se puede construir un ACA $(M; t, k, 3)$ agregando las filas necesarias a un ACA $(M; t, k, 2)$. Empíricamente se ha encontrado que es muy deseable que los alfabetos de una secuencia de ACA (uno de estos ACA se describe por $ACA_i(N_i; t, k, v_i), i = 0, 1, \dots$) pueda seguir la expresión: $v_i = 2^i + 1$, entonces los ACA que se utilizan son: $ACA_0(N_0; t, k, 2), ACA_1(N_1; t, k, 3), ACA_2(N_1; t, k, 5), ACA_3(N_1; t, k, 9), ACA_4(N_1; t, k, 17), ACA_5(N_1; t, k, 33), \dots, ACA_i(N_i; t, k, 2^i + 1)$ Para ilustrar que los ACA muestrean un número mayor de valores α que los CA, se generó el ACA $(42; 2; 10; 5)$ (construido basado en un ACA $(6; 2; 10; 2)$ y ACA $(16; 2; 10; 3)$) como se muestra en la **Tabla 3**, tenga en cuenta que los valores α se distribuyen en un rango más amplio, en este caso $3 \leq \alpha \leq 28$, con 15 valores α diferentes.

3.3 HIPOTESIS PRINCIPAL

La principal hipótesis que esta investigación prueba, es que al usar un ACA con N filas, valores bajos de fuerza ($t = \{2, 3\}$) y valores bajos del alfabeto ($v = \{9, 17\}$), el algoritmo MOEA/D-DE obtiene un mejor Frente de Pareto que el obtenido con N vectores de peso definidos con la estrategia actual (selección de los pesos que son más distintos entre sí con base en un pool amplio de vectores de peso generados aleatoriamente).

Tabla 3 ACA (42;2,10,5)

0	1	0	0	0	0	0	0	1	1	3
1	0	0	1	0	0	0	1	0	0	3
0	0	1	0	0	1	1	1	1	0	5
1	1	1	0	1	0	1	0	0	0	5
0	1	0	1	1	1	1	1	0	1	7
1	0	1	1	1	1	0	0	1	1	7
0	2	1	2	1	1	2	2	0	1	12
0	0	2	0	2	2	0	2	2	2	12
2	1	0	2	0	2	1	2	0	2	12
1	2	2	1	0	1	2	0	2	2	13
2	0	2	1	1	2	2	1	1	1	13
2	2	0	0	2	1	2	1	2	1	13
1	1	2	1	2	0	2	2	2	0	13
1	0	1	2	1	2	1	1	2	2	13
2	2	2	2	2	2	1	0	0	0	13
2	2	1	2	2	0	0	2	1	2	14
3	0	1	3	3	3	0	1	1	4	19
3	4	1	2	0	0	1	3	3	3	20
1	0	3	3	1	4	4	0	3	1	20
0	1	0	0	4	3	4	4	3	3	22
0	1	0	0	3	4	3	3	4	4	22
4	0	1	4	4	0	3	4	1	1	22
1	0	4	4	1	2	0	3	4	3	22
1	3	4	2	4	3	1	0	0	4	22
2	3	4	1	3	0	4	3	1	1	22
3	4	2	1	1	3	3	0	4	1	22
4	2	2	4	3	4	1	0	0	3	23
2	4	3	1	4	4	0	1	1	3	23
2	2	3	3	0	0	1	4	4	4	23
4	4	3	0	2	3	3	3	2	0	24
4	3	0	3	2	1	3	2	3	3	24
1	4	3	2	3	2	3	4	0	2	24
2	2	4	4	0	3	3	1	3	2	24
4	3	2	1	1	2	0	4	3	4	24
0	3	3	4	3	4	2	2	3	0	24
3	2	2	3	4	2	4	3	0	2	25
4	3	1	2	0	4	4	1	4	2	25
3	3	4	0	2	4	4	4	2	0	26
0	4	4	3	4	3	2	2	4	0	26
3	4	0	4	3	1	4	2	2	4	27
4	1	4	3	4	1	2	3	2	3	27
3	1	3	4	2	1	2	4	4	4	28

3.4 MOEA/D-DE-ACA

El **Algoritmo 3** presenta a MOEA/D-DE-ACA. Esta propuesta se centra en modificar el Paso 1 relacionado con la inicialización del muestreo aleatorio de los vectores de peso en MOEA/D-DE incorporando un ACA para obtener un muestro de tamaño fijo y mejor distribuido de los vectores de pesos. La propuesta mantiene los operadores genéticos del algoritmo MOEA/D-DE [12]. A continuación, se explica en detalle cada componente nuevo en MOEA/D-DE-ACA y los problemas solucionados al incorporar este método.

3.4.1 FUNDAMENTOS DE LA PROPUESTA

El algoritmo MOEA/D y MOEA/D-DE en su paso de inicialización, generan un muestreo aleatorio lo más uniformemente distribuido posible en el espacio de los objetivos para la definición de los vectores de peso de las soluciones en la población, esto implica generar m vectores de peso en los límites de los objetivos $(1, 0, 0, \dots, 0), (0, 1, 0, \dots, 0) \dots (0, 0, 0, \dots, 1)$ después generar una cantidad E ($E \gg N$) de vectores de peso candidatos, para luego hacer un proceso iterativo de selección de los vectores de peso candidatos que estén más alejados de la lista que ha sido previamente seleccionada e incluirlos hasta que se completen los N vectores de peso requeridos. Tarea que es computacionalmente costosa cuando el valor de E crece, proceso que tiene una complejidad de $O(N \times E)$.

Este método de inicialización permite a estos algoritmos que el usuario mantenga controlado el crecimiento en el número de vectores de peso que se usan para muestrear el espacio de objetivos a través del parámetro N (tamaño de la población). La calidad de ese muestreo se ve afectada por el crecimiento exponencial del número de vectores de peso que se deben tener en cuenta en el espacio de búsqueda de objetivos conforme se explicó en la sección anterior.

Con el uso de un ACA, la distribución de los vectores de peso permite evaluar de una mejor forma la interacción entre los objetivos. Además, el valor N (Numero de filas) del ACA permite controlar el tamaño de la población. Este valor N del ACA es definido por el número de objetivos k pero su crecimiento se controla con el valor del alfabeto v y la fuerza t . Además, los ACA se construyen fuera de línea o se adquieren y se pueden usar tantas veces como sea requerido, logrando con ello reducir el tiempo de ejecución del algoritmo de optimización de muchos objetivos.

En el Paso 1 del **Algoritmo 3** se toma fila por fila del ACA ($N, t, k = m, v$) y se realiza la conversión del dominio discreto de la fila del ACA al dominio continuo de los vectores de peso basado en el valor de α , lo que automáticamente permite obtener un vector de pesos normalizado. Esto se logra, sumando los valores de la fila y dividiendo cada valor de la fila por dicha suma. Proceso que se repite para las N filas del ACA y que permite obtener N vectores de peso normalizados.

A continuación, se muestra un ejemplo de esta conversión basado en la fila resaltada en la **Tabla 3** que corresponde a los siguientes valores **(3, 4, 2, 1, 1, 3, 3, 0, 4, 1)**. En este caso la suma de los valores en la fila es 22, que corresponde con el valor de α . Se dividen los valores individuales por la suma y se obtiene el siguiente vector de pesos **(0.14, 0.18, 0.09, 0.05, 0.05, 0.14, 0.14, 0.00, 0.18, 0.05)** que además esta normalizado conforme lo requiere el resto del algoritmo. En caso de que dos vectores de peso sean similares, se eliminan los duplicados y se reduce el valor de N al número de vectores de peso únicos. Aunque en espacios de 5 o más objetivos es muy poco probable que esta situación ocurra es necesario manejarla adecuadamente. En el siguiente ejemplo se ilustra la inicialización definida por ACA donde se toman lo objetivos 1 y objetivo 2, para mejor comprensión cada vector de peso w_i es normalizado y graficado en el espacio objetivo donde para el vector de peso w_1 en el objetivo 1 el valor es 3 después de normalizar el valor es 0,14, lo mismo pasa con el objetivo 2 su valor es 4 y normalizado pasa a ser 0,18, véase en la **Figura 14** como el vector de pesos w_1 toma como referencia los valores normalizados para el objetivo1 y el objetivo 2. Obviamente para los otros 8 objetivos que no se representan en la figura.

W	OBJ-1	OBJ-2	OBJ-3	OBJ-4	OBJ-5	OBJ-6	OBJ-7	OBJ-8	OBJ-9	OBJ-10	α
W_1	3	4	2	1	1	3	3	0	4	1	22
W_1	0,14	0,18	0,09	0,05	0,05	0,14	0,14	0,00	0,18	0,05	22
W_2	1	3	4	2	4	3	1	0	0	4	22
W_2	0,05	0,14	0,18	0,09	0,18	0,14	0,05	0,00	0,00	0,18	22
W_3	2	3	4	1	3	0	4	3	1	1	22
W_3	0,09	0,14	0,18	0,05	0,14	0,00	0,18	0,14	0,05	0,05	22
W_4	4	2	2	4	3	4	1	0	0	3	23
W_4	0,17	0,09	0,09	0,17	0,13	0,17	0,04	0,00	0,00	0,13	23
W_5	2	2	3	3	0	0	1	4	4	4	23
W_5	0,09	0,09	0,13	0,13	0,00	0,00	0,04	0,17	0,17	0,17	23

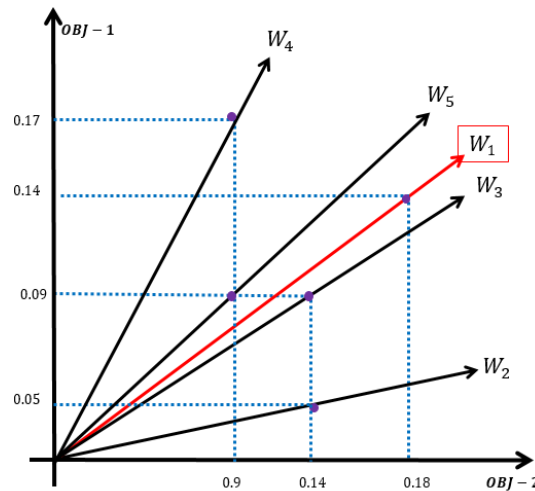


Figura 14 relación vector de peso en espacio objetivo

<p>Algoritmo 3 MOEA/D-DE-ACA</p>
<p>Entrada:</p> <ul style="list-style-type: none"> • Problema: Minimizar $F(x) = \text{Minimiza } (f_1(x), f_2(x), \dots, f_m(x))^T$, sujeto a $x \in \Omega$ donde $F: \Omega \rightarrow R^m$, consiste en m funciones objetivo de valor real presentes en el espacio objetivo, Ω es el espacio de decisión (variable), $\Omega = \{x \in R^n h_j(x) \leq 0, j = 1, \dots, m\}$ y h_j son funciones objetivo continuas • T: número de vectores de peso en el vecindario de cada vector de peso • δ: probabilidad de que las soluciones de los padres se seleccionen del vecindario • η_r: número máximo de soluciones remplazadas en cada generación. • ACA: Augmented Covering Array con N filas, fuerza t, m objetivos (o k columnas) y v alfabeto. CA $(N, t, k = m, v)$ • A Criterio de parada
<p>Salida: Aproximación al conjunto de Pareto (PS) : $\{x^1, \dots, x^N\}$ Aproximación al PF: $\{F(x^1), \dots, F(x^N)\}$</p>
<p>Inicialización</p> <ol style="list-style-type: none"> 1. Definir vectores de peso utilizando ACA. Defina los vectores de peso w^1, \dots, w^N basándose en la normalización de las filas ACA y actualice el valor de N si fuera necesario para eliminar las filas duplicadas. 2. Calcule las distancias euclidianas entre dos vectores de peso cualquiera y luego calcule los vectores de peso más cercanos a cada vector de peso. Para cada $i = 1, \dots, N$ establece $B(i) = \{i_1, \dots, i_T\}$ donde w^{i_1}, \dots, w^{i_T} son los T vectores de peso mas cercanos a w^i. 3. Genere una población inicial x^1, \dots, x^N aleatoria en Ω y Para cada $i = 1, \dots, N$ calcule $F(x^i)$. Inicialice $z = (z_1, \dots, z_m)$ donde $z_j = \min_{1 \leq i \leq N} f_j(x^i)$ 4. Actualización 5. Para $i = 1, \dots, N$ realice 6. Selección de rango de acoplamiento / actualización: Genere uniformemente un número de rand desde $[0, 1]$. Luego establece: $P = \begin{cases} B(i) & \text{if aleatorio} < \delta \\ \{1, \dots, N\} & \text{de lo contrario} \end{cases}$ 7. Reproducción: Conjunto $r_1 = i$ y seleccione aleatoriamente 2 índices r_2 y r_3 en P, genere una solución \bar{y} desde x^{r_1}, x^{r_2} y x^{r_3} por medio del operador DE, luego realice una mutación en \bar{y} con probabilidad p_m para producir una nueva solución y. 8. Reparación: Si un elemento de y esta fuera del limite, Ω, su valor se restablece para que sea un valor seleccionado al azar dentro del limite. Calcular $F(y)$. 9. Actualización de z: Para cada $j = 1, \dots, m$, if $f_j(y) < z_j$, establezca un conjunto $z_j = f_j(y)$. 10. Actualización de soluciones: establezca $c = 0$ y luego haga lo siguiente: 11. Mientras que P no está vacío hacer: <ol style="list-style-type: none"> a. Si $c = \eta_r$ vaya al Paso 13. b. Elegir aleatoriamente e indexar j de P. c. Si $g(y w^j, z) \leq g(x^j w^j, z)$ entonces establezca $x^j = y, F(x^j) = F(y)$, and $c = c + 1$. d. retire j de P. 12. Terminar Mientras 13. Criterio de parada: Si se cumple el criterio de parada, entonces detenga y envíe $\{x^1, \dots, x^N\}$ y $\{F(x^1), \dots, F(x^N)\}$. De lo contrario, dirigirse a Paso 6

Algoritmo 3 MOEA/D-DE-ACA

CAPÍTULO 4

4 EXPERIMENTACIÓN

4.1 PROBLEMAS USADOS PARA LA EVALUACIÓN Y COMPARACIÓN

Dos suites de prueba disponibles en MOEA Framework 2.12 fueron utilizados en este estudio, a saber: Deb-Thiele-Laumanns-Zitzler (DTLZ) y Walking-Fish-Group (WFG)[56]. Para cada problema de prueba, el número de objetivos varía de 10 a 100 en incrementos de 10, i.e. $m \in \{10, 20, 30, \dots, 100\}$. Todos los problemas pueden escalarse a cualquier número de objetivos y variables de decisión. La **Tabla 4** resume las propiedades de estos problemas.

En el dataset de pruebas DTLZ [56], el problema DTLZ1 tiene un PF lineal y regular, por lo que es fácil de resolver, DTLZ3, y DTLZ7, DTLZ2 presentan un PF convexo por lo que es bueno evaluar MOEAs con esta particularidad. DTLZ4 tiene una distribución no uniforme de puntos a lo largo del PF, y es bueno para evaluar MOEAs con capacidad para mantener una buena distribución de soluciones. En DTLZ5, su PF es una hipersuperficie degenerada. DTLZ6 tiene regiones óptimas desconectas en el PF y es bueno para evaluar la capacidad de los MOEA para mantener las subpoblaciones en partes desconectadas del espacio objetivo. El parámetro k para estos problemas se estableció en 5 para problemas DTLZ1, DTLZ5, y DTLZ6, 10 para los problemas DTLZ2, DTLZ3, y DTLZ4, y 20 para los problemas DTLZ5, donde el número de variables es $D = m + k - 1$.

En el dataset de prueba WFG, el problema WFG1 es unimodal separable, igual a WFG7, pero con diferentes formas del PF. Las formas del PF de los problemas WFG1, WFG2, y WFG3 son complicados, discontinuos y parcialmente degenerados, respectivamente. Hay cinco problemas WFG2, WFG3, WFG6, WFG8, y WFG9 que no son separables. WFG7, WFG8, y WFG9 están discontinuos y sesgados, pero WFG7 no es separable y WFG9 es un verdadero desafío debido a su alta modalidad. WFG4 como WFG9 también involucra multi-modalidad, pero no está sesgado y no es separable. WFG5 es más difícil que WFG9. El parámetro k para estos problemas se estableció igual a $m - 1$, y el parámetro de distancia l se estableció en 10, donde $D = k + l$.

Tabla 4 Propiedades de los problemas DTLZ y WFG usados para la evaluación donde $M = \{10, 20, 30, \dots, 100\}$

Problema	Forma de PF [57]	Multimodal [56]	Bias [56]	Discontinuo [56][57]	Separable [56]	Engañoso [56][57]	Escalar [57]	No. de Variables (D=M+k-1)	Generaciones
DTLZ1	Linear, Regular (Fácil)	Si [57]	No	No	Si * [58]	No		M+4 (k=5)	600
DTLZ2	Cóncavo	No	No	No	Si * [58]	No		M+9 (k=10)	500
DTLZ3	Cóncavo [59]	Si [57]	No	No	Si * [58]	Si		M+9 (k=10)	800
DTLZ4	Cóncavo [59]	No	Si	No	Si * [58]	No		M+9 (k=10)	500
DTLZ5	Cóncavo, Degenerado, Irregular	No	No	Desconocido	Desconocido	No		M+4 (k=5)	500
DTLZ6	Cóncavo, Degenerado, Irregular	No	Si	Desconocido	Desconocido	No		M+4 (k=5)	500
DTLZ7	Mixto	Si	Si [59]	Si [57]	No	Si	Si	M+19 (k=20)	500
WFG1	Convexo, Mixto	No	Si (Polynomial, Plano) [60]	No	Si	No	Si [61]	M+9 (≠10)	600
WFG2	Convexo [61] [60]	Si (F1:M-1 No) [60]	No	Si	No	No	Si [61]	M+9 (≠10)	500
WFG3	Linear, Degenerado	No	No	No	No	No	Si	M+9 (≠10)	500
WFG4	Cóncavo, Regular	Si (Altamente)	No	No	Si [59]	No	Si	M+9 (≠10)	500
WFG5	Cóncavo, Regular	No	No	No	Si	Si	Si	M+9 (≠10)	500
WFG6	Cóncavo, Regular	No	No	No	No	No	Si	M+9 (≠10)	500
WFG7	Cóncavo, Regular	No	Si (parámetro dependiente) [60]	No	Si	No	Si	M+9 (≠10)	500
WFG8	Cóncavo, Regular	No	Si (parámetro dependiente) [60]	No	No	No	Si	M+9 (≠10)	500
WFG9	Cóncavo, Regular	Si (Difícil)	Si (parámetro dependiente) [60]	No	No	Si	Si	M+9 (≠10)	500

4.2 MÉTRICA DE COMPARACIÓN

En este experimento, la distancia generacional invertida (DGI) [62] se adoptó para evaluar el rendimiento de los algoritmos comparados. La DGI es una métrica para evaluar la calidad del conjunto de soluciones obtenidas en términos de convergencia y diversidad. Entre 3.356 y 39.190 puntos muestreados uniformemente en los Frentes de Pareto se utilizan en el cálculo de la DGI para cada problema de prueba,

de acuerdo con el tamaño de la población. Un DGI más pequeño indica un mejor rendimiento. DGI se calcula con base en la **Ecuación 7** [63], donde P es un conjunto de aproximación, P^* es un conjunto de puntos no dominados y distribuidos uniformemente a lo largo del Frente de Pareto, $dist(z^* \in P)$ es la distancia euclidiana entre z^* y su vecino más cercano en P , y $|P^*|$ es la cardinalidad de P^* .

$$DGI(P) = \frac{1}{|P^*|} \sum_{z^* \in P^*} dist(z^* \in P)$$

Ecuación 7 Distancia Generacional Invertida

Las ventajas de DGI son dos principalmente: eficiencia computacional y generalidad. Si $|P^*|$ es lo suficientemente grande para cubrir muy bien el verdadero PF, entonces tanto la convergencia como la diversidad de P se puede medir por DGI. Varios artículos señalan que la DGI es la métrica más apropiada para evaluar MOEAs que resuelven MaOPs.

4.3 AJUSTE DE PARAMETROS

En nuestro estudio, el algoritmo propuesto se compara con MOEA/D-DE y NSGA-III. Los parámetros de configuración de los algoritmos se fijaron según lo recomendado en la literatura, los cuales se resumen en la **Tabla 5**, donde n es la variable dimensión del espacio de búsqueda, p_m y p_c son las probabilidades de mutación y cruce respectivamente, η_m y η_c son los índices de distribución de los operadores de cruce y mutación respectivamente, T es el tamaño del vecindario de los vectores de peso, n_r es el número máximo de soluciones reemplazadas por cada nueva solución, y δ es la probabilidad de seleccionar soluciones de padres en el vecindario. El tamaño de la población N se definió para todos los algoritmos en función del tamaño del ACA seleccionado para el problema específico. El número máximo de evaluaciones de funciones (Max_FEs) es el criterio de detención para todos los algoritmos y es el resultado del parámetro número de generaciones (columna final, **Tabla 6**) multiplicado por el parámetro del tamaño de la población.

Tabla 5 Ajustes de parámetros para los algoritmos comparados

Algoritmos	Ajuste de parámetros
NSGA-III	$p_c = 1.0, p_m = 1/n, \eta_c = 30, \eta_m = 20$
MOEA/D-DE	$p_m = 1/n, \eta_m = 20, CR = 1, F = 0.5, \delta = 0.9, T = 20, n_r = 2$
MOEA/D-DE-ACA	$p_m = 1/n, \eta_m = 20, CR = 1, F = 0.5, \delta = 0.9, T = 20, n_r = 2$

4.4 RESULTADOS DE MOEA/D-DE-ACA CON $v=9$ $t=2$

A continuación, en la **Tabla 6** y en la **Tabla 7** se presentan los resultados promedio de DGI de las 31 repeticiones para los tres algoritmos (MOEA/D-DE-ACA, MOEA/D-DE y NSGA-III) usando 10, 20, ..., hasta 100 objetivos en los siete problemas DTLZ y nueve problemas WFG, respectivamente. Las celdas con negrilla corresponden al mejor resultado en cada experimento y las que además tienen fondo gris son aquellas logradas por MOEA/D-DE-ACA. Cada celda muestra el puesto (ranking) del algoritmo entre paréntesis y el valor promedio de DGI logrado por el algoritmo con el número de objetivos establecido en la fila sobre el problema definido en la columna.

En la **Tabla 6** se puede observar lo siguiente:

- 1) Sin importar el número de objetivos (10 a 100), MOEA/D-DE-ACA obtiene mejores valores promedio de DGI en los problemas DTLZ1, DTLZ2 y DTLZ3, que son problemas lineales (el primero) y cóncavos (los dos siguientes), no tienen sesgo (bias), no tienen discontinuidades y son separables (contando una excepción con 40 objetivos en DTLZ1 donde es superado por MOEA/D-DE en 0.0039).
- 2) En el problema DTLZ7, MOEA/D-DE-ACA obtiene los mejores resultados de DGI desde los 10 hasta los 70 objetivos, siendo este un problema con un frente de Pareto mixto, multimodal, desconectado, no separable y desactivado.
- 3) MOEA/D-DE-ACA obtiene el mejor DGI para el problema DTLZ6 desde los 50 hasta los 100 objetivos y para el problema DTLZ5 desde los 70 objetivos hasta los 100 objetivos. Problemas que tienen un frente de Pareto cóncavo, degenerado e irregular, que no son multimodales y que no son discontinuos. En los restantes objetivos (10 a 60 en DTLZ5 y 20 a 40 en DTLZ6) ocupa el segundo puesto siendo superado por MOEA/D-DE (entre 0.00021 y 0.00099).
- 4) Finalmente, en el problema DTLZ4, MOEA/D-DE-ACA solo gana con 10 y 30 objetivos, en los otros objetivos sólo es superado por MOEA/D-DE y las diferencias son menores a 13 décimas. Este problema, a pesar de tener características similares a DTLZ2 tiene bias.
- 5) El algoritmo NSGA-III en general es último en todos los rankings y las distancias de los valores promedio de DGI que obtiene se hacen más distantes de los obtenidos por los otros dos algoritmos en la medida que el número de objetivos aumenta, lo que muestra una importante debilidad para el uso de este algoritmo con problemas DTLZ con un número alto de objetivos.
- 6) MOEA/D-DE obtiene el primer puesto en DTLZ6 entre 20 y 40 objetivos, en DTLZ5 entre 10 y 60 objetivos y en DTLZ4 entre 40 y 100 objetivos, ganando en

DTLZ5 y DTLZ6 por márgenes estrechos (milésimas), pero en DTLZ4 por un margen mayor, decimas.

Tabla 6 Resultados de evaluación en problemas DTLZ con ACA $v=9$ $t=2$

ALGORITMOS	M	OBJ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEA/D-DE-ACA			(1) 0.1457	(1) 0.3194	(1) 0.1888	(1) 0.2919	(2) 0.1505	(1) 0.1044	(1) 0.6671
MOEA/D-DE	DGI	10	(2) 0.1513	(2) 0.3582	(2) 0.2033	(3) 0.3163	(1) 0.1481	(2) 0.1046	(2) 0.7505
NSGA-III			(3) 0.1696	(3) 0.3746	(3) 0.2989	(2) 0.3024	(3) 0.2172	(3) 0.5621	(3) 1.1480
MOEA/D-DE-ACA			(1) 0.1232	(1) 0.4372	(1) 0.1411	(2) 0.3697	(2) 0.2045	(2) 0.1279	(1) 1.1480
MOEA/D-DE	DGI	20	(2) 0.1364	(3) 0.4820	(2) 0.1471	(3) 0.4221	(1) 0.2024	(1) 0.1179	(2) 1.5285
NSGA-III			(3) 0.1612	(2) 0.4458	(3) 0.4070	(1) 0.3295	(3) 0.2341	(3) 0.4297	(3) 1.7975
MOEA/D-DE-ACA			(1) 0.1245	(1) 0.4905	(1) 0.1647	(1) 0.4614	(2) 0.2010	(2) 0.1049	(1) 1.7280
MOEA/D-DE	DGI	30	(2) 0.1423	(2) 0.5359	(2) 0.1673	(2) 0.4825	(1) 0.1939	(1) 0.0992	(2) 2.0447
NSGA-III			(3) 0.2119	(3) 0.6693	(3) 0.3213	(3) 0.4966	(3) 0.2951	(3) 0.4388	(3) 2.2949
MOEA/D-DE-ACA			(2) 0.1039	(1) 0.5235	(1) 0.1571	(2) 0.5359	(2) 0.1618	(2) 0.0912	(1) 2.2291
MOEA/D-DE	DGI	40	(1) 0.1000	(2) 0.5718	(2) 0.1673	(1) 0.5092	(1) 0.1590	(1) 0.0906	(2) 2.5549
NSGA-III			(3) 0.1758	(3) 0.7721	(3) 0.3086	(3) 0.5871	(3) 0.2965	(3) 0.4688	(3) 2.6981
MOEA/D-DE-ACA			(1) 0.0663	(1) 0.5547	(1) 0.1250	(2) 0.6354	(2) 0.1869	(1) 0.0887	(1) 2.6861
MOEA/D-DE	DGI	50	(2) 0.0698	(2) 0.6165	(2) 0.1337	(1) 0.5506	(1) 0.1846	(2) 0.0903	(2) 2.9569
NSGA-III			(3) 0.1169	(3) 0.8753	(3) 0.3301	(3) 0.6740	(3) 0.2929	(3) 0.4995	(3) 3.0068
MOEA/D-DE-ACA			(1) 0.0615	(1) 0.5812	(1) 0.1226	(2) 0.6891	(2) 0.1816	(1) 0.0722	(1) 3.0971
MOEA/D-DE	DGI	60	(2) 0.0746	(2) 0.6349	(2) 0.1298	(1) 0.5763	(1) 0.1800	(2) 0.0759	(2) 3.2730
NSGA-III			(3) 0.1325	(3) 0.9522	(3) 0.4146	(3) 0.7245	(3) 0.2877	(3) 0.5283	(3) 3.3230
MOEA/D-DE-ACA			(1) 0.0835	(1) 0.6065	(1) 0.1230	(2) 0.7266	(1) 0.1807	(1) 0.0670	(1) 3.4231
MOEA/D-DE	DGI	70	(2) 0.0926	(2) 0.6516	(2) 0.1285	(1) 0.6016	(2) 0.1809	(2) 0.0716	(3) 3.6111
NSGA-III			(3) 0.1425	(3) 0.9849	(3) 0.5539	(3) 0.7859	(3) 0.2934	(3) 0.6775	(2) 3.5931
MOEA/D-DE-ACA			(1) 0.0906	(1) 0.6148	(1) 0.1329	(2) 0.7491	(1) 0.1861	(1) 0.0626	(2) 3.7727
MOEA/D-DE	DGI	80	(2) 0.0947	(2) 0.6498	(2) 0.1458	(1) 0.6209	(2) 0.1891	(2) 0.0725	(3) 3.9323
NSGA-III			(3) 0.1614	(3) 0.9839	(3) 0.6074	(3) 0.8184	(3) 0.2952	(3) 1.1399	(1) 3.7696
MOEA/D-DE-ACA			(1) 0.0914	(1) 0.6224	(1) 0.1444	(2) 0.7710	(1) 0.1981	(1) 0.0824	(2) 4.1787
MOEA/D-DE	DGI	90	(2) 0.1018	(2) 0.6635	(2) 0.1596	(1) 0.6381	(2) 0.2027	(2) 0.0977	(3) 4.2406
NSGA-III			(3) 0.1750	(3) 1.0005	(3) 0.7430	(3) 0.8398	(3) 0.3099	(3) 4.2159	(1) 4.0912
MOEA/D-DE-ACA			(1) 0.1063	(1) 0.6304	(1) 0.1342	(2) 0.7967	(1) 0.1719	(1) 0.0824	(3) 4.5781
MOEA/D-DE	DGI	100	(2) 0.1160	(2) 0.6758	(2) 0.1454	(1) 0.6692	(2) 0.1757	(2) 0.0977	(2) 4.5343
NSGA-III			(3) 0.1994	(3) 1.0274	(3) 0.6672	(3) 0.8838	(3) 0.3037	(3) 4.2159	(1) 4.2818

En la **Tabla 7** se puede observar lo siguiente:

1) En los problemas WFG4, WFG5, WFG6 y WFG8 desde los 30 hasta los 100 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI. Estos problemas se caracterizan por tener un frente de Pareto cóncavo, regular, y ser continuos.

2) En el problema WFG2 desde los 10 hasta los 60 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI seguido por MOEA/D-DE con diferencias que aumentan poco a poco. Este problema es convexo, multimodal, desconectado y escalable.

3) En el problema WFG7 desde los 40 hasta los 70 objetivos, MOEA/D-DE-ACA obtiene mejores valores de DGI seguido por MOEA/D-DE con diferencias que disminuyen poco a poco. Este problema es similar a WFG8 donde el algoritmo domina de 30 a 100 objetivos, pero es separable, hecho que le ayuda a tener mejores resultados en este problema.

4) El algoritmo NSGA-III tiene valores óptimos en cantidad baja de objetivos (10 y 20 en problemas WFG4 a WFG9), pero a medida que crece el número de objetivos los valores tienden a subir (desmejorar). A pesar de eso, para los problemas WFG1 y WFG3, NSGA-III domina de 20 a 100 objetivos con diferencias importantes en valores promedio de DGI.

5) MOEA/D-DE obtienen en general valores intermedios de DGI en estos problemas y logra un dominio en el problema WFG9 a partir de los 60 objetivos con un frente de Pareto cóncavo y regular. En este sentido la inicialización de los pesos con ACAs le permite a MOEA/D-DE mejorar su desempeño en los problemas WFG con frentes de Pareto irregulares, discontinuos y que se escalan en diferentes rangos.

Tabla 7 Resultados de evaluación en problemas WFG con ACA $v=9$ $t=2$

ALGORITMOS	M	OBJ	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
MOEA/D-DE-ACA			(2) 0.1089	(1) 0.1059	(1) 0.1792	(3) 0.4017	(3) 0.3018	(3) 0.4681	(3) 0.3453	(3) 0.4983	(3) 0.3782
MOEA/D-DE	DGI	10	(1) 0.0861	(2) 0.1127	(3) 0.2074	(2) 0.3719	(2) 0.2834	(2) 0.4176	(2) 0.3440	(2) 0.4603	(2) 0.3460
NSGA-III			(3) 0.1779	(3) 0.1676	(2) 0.1836	(1) 0.3441	(1) 0.2755	(1) 0.3651	(1) 0.3213	(1) 0.3808	(1) 0.2964
MOEA/D-DE-ACA			(2) 3.3927	(1) 0.1268	(3) 0.2264	(3) 0.4800	(2) 0.3958	(2) 0.5574	(2) 0.4782	(2) 0.5703	(2) 0.4503
MOEA/D-DE	DGI	20	(3) 4.2079	(2) 0.1507	(2) 0.2218	(2) 0.4764	(3) 0.4662	(3) 0.5858	(3) 0.5280	(3) 0.5747	(3) 0.4892
NSGA-III			(1) 0.9033	(3) 0.2007	(1) 0.2103	(1) 0.3963	(1) 0.3272	(1) 0.3942	(1) 0.3664	(1) 0.4135	(1) 0.3443
MOEA/D-DE-ACA			(2) 4.1126	(1) 0.1564	(3) 0.2377	(1) 0.5587	(1) 0.4487	(1) 0.6216	(2) 0.5629	(1) 0.6175	(2) 0.5107
MOEA/D-DE	DGI	30	(3) 6.4187	(2) 0.1921	(2) 0.2302	(2) 0.6130	(2) 0.4825	(3) 0.6427	(3) 0.5859	(3) 0.6342	(3) 0.5352
NSGA-III			(1) 1.0403	(3) 0.2431	(1) 0.2135	(3) 0.6403	(3) 0.5015	(2) 0.6354	(1) 0.5171	(2) 0.6301	(1) 0.5049
MOEA/D-DE-ACA			(2) 9.8444	(1) 0.1768	(3) 0.2669	(1) 0.5783	(1) 0.4802	(1) 0.6320	(1) 0.5886	(1) 0.6111	(1) 0.6022
MOEA/D-DE	DGI	40	(3) 11.751	(2) 0.2150	(2) 0.2587	(2) 0.6321	(2) 0.5115	(2) 0.6711	(3) 0.6386	(2) 0.6433	(2) 0.6168
NSGA-III			(1) 2.0611	(3) 0.2657	(1) 0.2259	(3) 0.7516	(3) 0.5776	(3) 0.7555	(2) 0.6133	(3) 0.7156	(3) 0.7088
MOEA/D-DE-ACA			(3) 15.400	(1) 0.1896	(2) 0.2748	(1) 0.6079	(1) 0.4979	(1) 0.6621	(1) 0.6270	(1) 0.6438	(1) 0.6390
MOEA/D-DE	DGI	50	(2) 14.295	(2) 0.2214	(3) 0.2755	(2) 0.6790	(2) 0.5326	(2) 0.7115	(2) 0.6554	(2) 0.6972	(2) 0.6410
NSGA-III			(1) 5.3195	(3) 0.2760	(1) 0.2237	(3) 0.8358	(3) 0.6319	(3) 0.8348	(3) 0.6837	(3) 0.8072	(3) 0.8069
MOEA/D-DE-ACA			(3) 9.6503	(1) 0.2472	(3) 0.3087	(1) 0.6349	(1) 0.5237	(1) 0.6884	(1) 0.6406	(1) 0.6806	(2) 0.6610
MOEA/D-DE	DGI	60	(2) 7.1777	(2) 0.2564	(2) 0.2868	(2) 0.6899	(2) 0.5406	(2) 0.7334	(3) 0.6524	(2) 0.7288	(1) 0.6473
NSGA-III			(1) 4.1623	(3) 0.3080	(1) 0.2416	(3) 0.8932	(3) 0.6802	(3) 0.9006	(2) 0.6412	(3) 0.9166	(3) 0.8777
MOEA/D-DE-ACA			(3) 43.702	(2) 0.2905	(3) 0.3317	(1) 0.6496	(1) 0.5407	(1) 0.7016	(1) 0.6643	(1) 0.7013	(2) 0.6953
MOEA/D-DE	DGI	70	(2) 26.651	(1) 0.2819	(2) 0.3141	(2) 0.7093	(2) 0.5579	(2) 0.7560	(3) 0.6888	(2) 0.7459	(1) 0.6577
NSGA-III			(1) 22.747	(3) 0.3266	(1) 0.2290	(3) 0.9360	(3) 0.6958	(3) 0.9388	(2) 0.6692	(3) 0.9770	(3) 0.9486
MOEA/D-DE-ACA			(3) 4.9379	(3) 0.3527	(3) 0.3534	(1) 0.6500	(1) 0.5574	(1) 0.7168	(3) 0.6902	(1) 0.7118	(2) 0.7222
MOEA/D-DE	DGI	80	(2) 2.9548	(1) 0.2980	(2) 0.3347	(2) 0.7014	(2) 0.5698	(2) 0.7562	(1) 0.6756	(2) 0.7452	(1) 0.6742
NSGA-III			(1) 2.5396	(2) 0.3512	(1) 0.2236	(3) 0.9627	(3) 0.7295	(3) 0.9944	(2) 0.6833	(3) 1.0088	(3) 1.0464
MOEA/D-DE-ACA			(2) 3.0815	(2) 0.3612	(3) 0.3428	(1) 0.6653	(1) 0.5692	(1) 0.7177	(2) 0.7226	(1) 0.7056	(2) 0.7461
MOEA/D-DE	DGI	90	(3) 3.2389	(1) 0.3103	(2) 0.3205	(2) 0.7182	(2) 0.5837	(2) 0.7641	(1) 0.7002	(2) 0.7415	(1) 0.6669
NSGA-III			(1) 2.0892	(3) 0.3766	(1) 0.2394	(3) 0.9852	(3) 0.7565	(3) 1.0150	(3) 0.7403	(3) 1.0465	(3) 1.0802
MOEA/D-DE-ACA			(2) 3.5191	(3) 0.3960	(3) 0.3538	(1) 0.6602	(1) 0.5731	(1) 0.7161	(3) 0.7167	(1) 0.6930	(2) 0.7771
MOEA/D-DE	DGI	100	(3) 3.6717	(1) 0.3455	(2) 0.3502	(2) 0.7234	(2) 0.5907	(2) 0.7739	(1) 0.7032	(2) 0.7473	(1) 0.6738
NSGA-III			(1) 1.2328	(2) 0.3807	(1) 0.2465	(3) 1.0142	(3) 0.7612	(3) 1.0453	(2) 0.7032	(3) 1.0746	(3) 1.1486

4.4.1 Test de Friedman para experimentos con $t=2$ y $v=9$

Partiendo de que el análisis previo dificulta determinar con exactitud cuál algoritmo es mejor y en qué tipo de problemas, Se procede a evaluar el comportamiento promedio de los tres algoritmos en estos problemas usando la prueba no paramétrica de Friedman con el post hoc de Holm. Esta prueba se realizó con 2 grados de libertad y los resultados de Holm se evaluaron con un nivel de significancia del 90% y 95%.

La **Tabla 8** muestra en la primera columna el número de objetivos (10 a 100) evaluados en los problemas DTLZ y WFG, luego los tres algoritmos con su ranking (1, 2 o 3) y el ranking de Friedman, después el valor p obtenido en la prueba y si dicho valor es significativo o no (True y False).

Luego, se observa el resultado del post hoc de Holm con una matriz de 3x3, donde la primera fila y columna se refiere al algoritmo A (MOEA/D-DE-ACA), la segunda fila y columna al algoritmo B (MOEA/D-DE) y la tercera fila y columna al algoritmo C (NSGA-III). El símbolo ● expresa que los resultados obtenidos con el algoritmo de la fila dominan a los resultados obtenidos con el algoritmo de la columna, mientras que el símbolo ○ expresa que el algoritmo de la columna domina al de la fila. Los valores por encima de la diagonal tienen un nivel de significancia del 90%, mientras que los ubicados debajo de la diagonal tienen una significancia del 95%. Después aparece el tamaño de la población con la que se ejecutaron los tres algoritmos, valor definido por el ACA tomado en MOEA/D-DE-ACA y que tiene la fortaleza de crecer logarítmicamente en función del número de objetivos (Tamaño de población = $1681.0 \ln(\text{número de objetivos}) - 408.78$ con R^2 de 0.973). La última columna de esta tabla muestra el número de puntos de referencia usados por MOEA Framework para el cálculo de DGI en los problemas, de acuerdo con el número de objetivos, en este caso de 3356 para 10 objetivos y de 7416 para 100 objetivos.

En la **Tabla 8** se puede apreciar que el algoritmo MOEA/D-DE-ACA obtiene en todos los casos el ranking número 1, pero en 10 y 20 objetivos, este ranking no es estadísticamente significativo (el valor p obtenido no es menor que 0.05), razón por la cual para estos dos experimentos no se realiza el post hoc de Holm. Basado en Holm, se puede observar que, entre 30 y 60 objetivos, MOEA/D-DE-ACA domina a MOEA/D-DE y a NSGA-III con un nivel de significancia del 95%. Entre 70 y 100 objetivos no se puede establecer una relación de dominancia entre MOEA/D-DE-ACA y MOEA/D-DE, pero estos dos algoritmos dominan a NSGA-III con un 90% de significancia. Esta misma relación se puede definir con un 95% de significancia con 70 y 90 objetivos. Con 90 objetivos MOEA/D-DE-ACA también domina a MOEA/D-DE con 95% de significancia.

Tabla 8 Resultados test de Friedman y post hoc de Holm en DGI para fuerza ($t=2$) y alfabeto ($v=9$)

Obj.	MOEA/D-DE-ACA (A)	MOEA/D-DE (B)	NSGA-III (C)	p-value	Significancia	Holm	Tamaño de población	Puntos de referencia																
10	(1)	1.88	(2) 2.00	(3) 2.13	0.7788 0	False	-	136 3356																
20	(1)	1.81	(3) 2.38	(1) 1.81	0.1849 8	False	-	174 4830																
30	(1)	1.44	(2) 2.19	(3) 2.38	0.0195 0	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>-</td></tr> <tr><td>C</td><td>○</td><td>-</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	-	C	○	-	-	197 5414
	A	B	C																					
A	-	●	●																					
B	○	-	-																					
C	○	-	-																					
40	(1)	1.44	(2) 1.88	(3) 2.69	0.0016 0	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	●	C	○	○	-	215 5562
	A	B	C																					
A	-	●	●																					
B	○	-	●																					
C	○	○	-																					
50	(1)	1.31	(2) 1.94	(3) 2.75	0.0002 5	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	●	C	○	○	-	232 6303
	A	B	C																					
A	-	●	●																					
B	○	-	●																					
C	○	○	-																					
60	(1)	1.44	(2) 1.88	(3) 2.69	0.0016 0	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	●	C	○	○	-	245 6576
	A	B	C																					
A	-	●	●																					
B	○	-	●																					
C	○	○	-																					
70	(1)	1.44	(2) 1.94	(3) 2.63	0.0033 9	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>B</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	-	●	B	-	-	●	C	○	○	-	256 6390
	A	B	C																					
A	-	-	●																					
B	-	-	●																					
C	○	○	-																					
80	(1)	1.69	(2) 1.81	(3) 2.50	0.0467 7	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>B</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>-</td><td>-</td></tr> </table>		A	B	C	A	-	-	●	B	-	-	●	C	○	-	-	266 6753
	A	B	C																					
A	-	-	●																					
B	-	-	●																					
C	○	-	-																					
90	(1)	1.50	(2) 1.88	(3) 2.63	0.0052 5	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	-	-	●	C	○	○	-	277 7416
	A	B	C																					
A	-	●	●																					
B	-	-	●																					
C	○	○	-																					
100	(1)	1.69	(2) 1.81	(3) 2.50	0.0467 7	True	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>B</td><td>-</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>-</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	-	●	B	-	-	●	C	-	○	-	288 7416
	A	B	C																					
A	-	-	●																					
B	-	-	●																					
C	-	○	-																					

4.4.2 Tiempo de ejecución para experimentos con $t=2$ y $v=9$

La **Figura 11** muestra el tiempo promedio de ejecución (Average Execution Time, AET) de los tres algoritmos para todos los datasets DTLZ y WFG con cada número de objetivos, de 10 a 100. Se puede apreciar que el tiempo de ejecución de NSGA-III es mucho mayor que el de los otros dos algoritmos, en este caso el crecimiento es cuadrático ($AET = 213,1 objectives^2 - 10533 objectives + 121769$ con un $R^2 = 0.9966$). Esto muestra que el procesamiento ejecutado por NSGA-III más allá de las evaluaciones de la función objetivo es mucho mayor al realizado por los otros dos algoritmos, ya que todos ejecutan el mismo número de EFOs. Los tiempos de ejecución de MOEA/D-DE ($AET = 8,3186 objectives^2 + 245,9 objectives -$

2895,5 con un $R^2 = 0.9953$) y MOEA/D-DE-ACA ($AET = 3,6936 objectives^2 + 63,126 objectives - 397,44$ con un $R^2 = 0.998$) también tienen una tendencia cuadrática donde MOEA/D-DE-ACA es el que presenta el menor tiempo de ejecución (menor constante de multiplicación). La diferencia obtenida en el tiempo entre las dos versiones de MOEA/D se explica en el tiempo que se ahorra en el Paso 1.1 de Inicialización; el uso de un ACA previamente fabricado da una ventaja en tiempo con el enfoque propuesto.

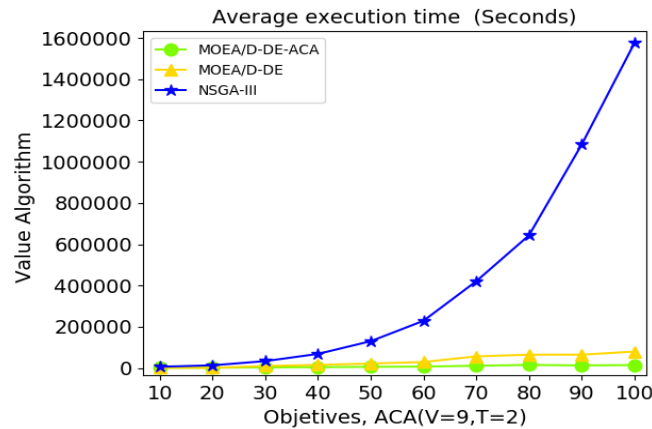


Figura 11 Tiempo promedio de ejecución en segundos para todos los algoritmos en los experimentos con fuerza 2 y alfabeto 9

4.4.3 Resultados por problema para experimentos con $t=2$ y $v=9$

La **Figura 12** muestra un gráfico box-plot que resume visualmente la mediana y los cuartiles de los valores DGI obtenidos por los tres algoritmos en los 16 problemas (DTLZ y WFG) en los experimentos con $t=2$ y $v=9$ de 10 a 100 objetivos. Se puede observar que, en 12 problemas (DTLZ1 a DTLZ3, DTLZ5 a DTLZ7, WFG2 y WFG4 a WFG8) que corresponde al 75% de los 16 problemas, MOEA/D-DE-ACA obtiene mejores (menores) valores medianos de DGI que los otros dos algoritmos, seguido por MOEA/D-DE.

En DTLZ4 y WFG9, MOEA/D-DE-ACA es superado por MOEA/D-DE y en WFG1 y WFG3, los mejores resultados los obtiene NSGA-III dejando en el primer problema a MOEA/D-DE-ACA en segundo lugar y en el otro de tercero.

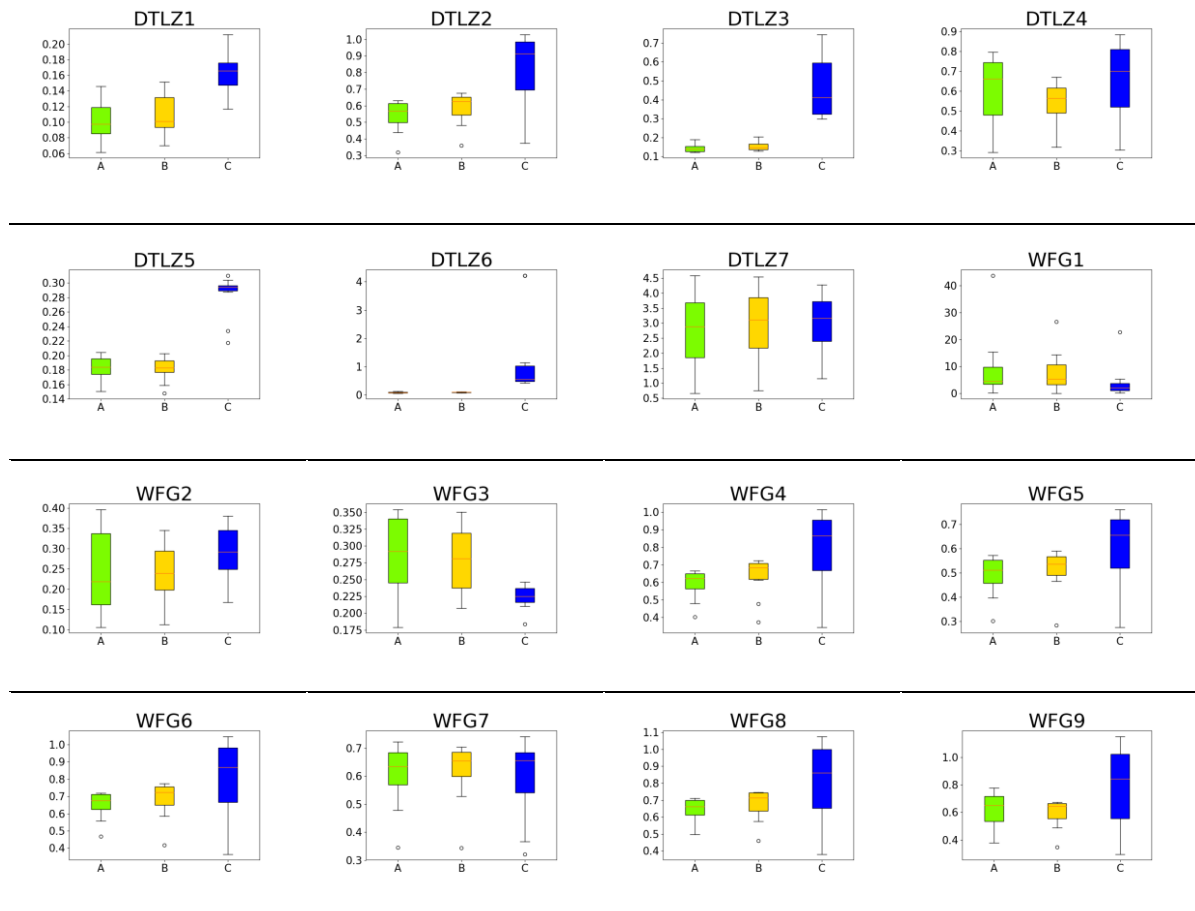


Figura 12 . Valores de DGI obtenidos para todos los algoritmos en la evaluación de $v = 9$ y $t = 2$ de 10 a 100 objetivos. A: MOEA / D-DE-ACA, B: MOEA / D-DE y C: NSGA-III

4.1 EXPERIMENTOS CON FUERZA $t=2$ Y ALFABETO $v=17$

En relación con el experimento anterior, se cambia el valor del alfabeto de 9 a 17 (casi el doble), lo que implica que los ACAs tengan más renglones. Con 10 objetivos se pasa de 136 filas en $v=9$ a 416 con $v=17$, que establece una relación de 1 a 3.06, mientras que con 100 objetivos se pasa de 288 con $v=9$ a 838 con $v=17$, que es una relación de 1 a 2.91. En las relaciones intermedias para los otros objetivos se obtiene un máximo de 1 a 3.45. Esto muestra en este caso que duplicar el alfabeto triplica (aproximadamente) el número de renglones (filas) en el ACA y con ello el número de vectores de peso en MOEA/D- DE-ACA.

En la **Tabla 9** se puede observar lo siguiente:

- 1) MOEA/D-DE-ACA obtiene mejores valores promedio desde 40 hasta 100 objetivos para DGI en todos los problemas DTLZ con cinco excepciones donde es segundo (diferencia máxima de 0.0316) y en DTLZ4 donde es superado por los otros dos algoritmos (diferencia máxima de 0,149).

- 2) En los problemas DTLZ3 y DTLZ7, MOEA/D-DE-ACA obtiene los mejores resultados de DGI desde los 20 objetivos, teniendo estos problemas un frente de Pareto cóncavo, el primero de ellos mixto, el segundo multimodales, no separable, se observa que MOEA/D-DE-ACA mejora el rendimiento en estos dos problemas cuando el número objetivos crece hasta 60-70 y luego se mantiene hasta los 100 objetivos.
- 3) MOEA/D-DE-ACA en los objetivos 10 a 30 de los problemas DTLZ5 y DTLZ6 ocupa el segundo puesto siendo superado por MOEA/D-DE (diferencias entre 0.0005 y 0.0053).
- 4) El algoritmo NSGA-III en general lidera los rankings de los valores promedio de DGI de 10 y 20 objetivos de los problemas DTLZ1 a DTLZ4, pero después sus valores son superados por los obtenidos por los otros dos algoritmos en la medida que el número de objetivos aumenta, lo que muestra una importante debilidad para el uso de este algoritmo con problemas DTLZ con un número alto de objetivos.
- 5) Similar al experimento anterior, MOEA/D-DE en general obtiene el segundo puesto salvo unas excepciones donde la mayoría de ellas se ubican de 10 a 40 objetivos.

En la **Tabla 10** se puede observar lo siguiente:

- 1) En el problema WFG2 desde los 20 hasta los 100 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI, seguido por MOEA/D-DE con diferencias que aumentan poco a poco a medida que incrementa el número de objetivos.
- 2) En los problemas WF4 y WF6 desde los 50 hasta los 100 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI, estos problemas se caracterizan por tener un frente de Pareto cóncavo, regular, no tener bias, no ser discontinuos, y ser escalables, pero el primero es altamente multimodal y el segundo no.
- 3) En los problemas WF5 y WF8 desde los 60 hasta los 100 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI, estos problemas se caracterizan por tener un frente de Pareto cóncavo, regular, no ser multimodales, no ser desconectado, el primero es separable y el segundo no, el primero es discontinuos pero el segundo no y ser escalable.
- 4) En general, el algoritmo NSGA-III ocupa el primer lugar en los problemas WFG1 excepto WFG2 desde 10 hasta 40 objetivos, pero similar con más objetivos, el valor promedio de DGI para este algoritmo se hace más grande (se aleja del PF ideal). A pesar de esto en los problemas WFG3 y WFG7 obtiene los mejores resultados hasta los 100 objetivos y obtiene los mejores resultados en WFG1 con 90 y 100 objetivos.
- 5) MOEA/D-DE obtiene los mejores resultados de DGI en WFG9 de 60 a 100 objetivos seguido muy de cerca por MOEA/D-DE-ACA.

Tabla 9 Resultados de evaluación en problemas DTLZ con ACA $v=17$ $t=2$

ALGORITMOS	M	OBJ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEA/D-DE-ACA			(3) 0.1340	(3) 0.2535	(3) 0.2520	(3) 0.2076	(2) 0.1308	(2) 0.0719	(3) 0.7325
MOEA/D-DE	DGI	10	(2) 0.1261	(2) 0.2468	(2) 0.2429	(2) 0.1917	(1) 0.1255	(1) 0.0689	(1) 0.5904
NSGA-III			(1) 0.0952	(1) 0.1714	(1) 0.1737	(1) 0.1209	(3) 0.1437	(3) 0.4932	(2) 0.6666
MOEA/D-DE-ACA			(2) 0.1791	(2) 0.3085	(1) 0.2484	(2) 0.2409	(2) 0.1903	(2) 0.0839	(1) 0.9683
MOEA/D-DE	DGI	20	(3) 0.1937	(3) 0.3535	(3) 0.2983	(3) 0.2508	(1) 0.1860	(1) 0.0818	(2) 1.2285
NSGA-III			(1) 0.1402	(1) 0.2761	(2) 0.2527	(1) 0.1508	(3) 0.2043	(3) 0.3201	(3) 1.8542
MOEA/D-DE-ACA			(2) 0.1304	(2) 0.3631	(1) 0.2536	(2) 0.2815	(2) 0.1845	(2) 0.0667	(1) 1.5502
MOEA/D-DE	DGI	30	(1) 0.1298	(3) 0.4060	(3) 0.2898	(3) 0.3453	(1) 0.1831	(1) 0.0662	(2) 1.7989
NSGA-III			(3) 0.1669	(1) 0.2770	(2) 0.2550	(1) 0.1457	(3) 0.2673	(3) 0.4951	(3) 2.4552
MOEA/D-DE-ACA			(1) 0.0833	(2) 0.4284	(1) 0.2003	(2) 0.3575	(2) 0.1927	(1) 0.0518	(1) 2.1349
MOEA/D-DE	DGI	40	(2) 0.0836	(3) 0.4653	(2) 0.2215	(3) 0.4026	(1) 0.1908	(2) 0.0541	(2) 2.2939
NSGA-III			(3) 0.1174	(1) 0.3968	(3) 0.3968	(1) 0.2797	(3) 0.2645	(3) 0.4922	(3) 2.8666
MOEA/D-DE-ACA			(1) 0.0894	(1) 0.4690	(1) 0.1589	(2) 0.4079	(2) 0.1788	(2) 0.0498	(1) 2.5091
MOEA/D-DE	DGI	50	(2) 0.1002	(2) 0.5189	(2) 0.1936	(3) 0.4400	(1) 0.1768	(1) 0.0493	(2) 2.7684
NSGA-III			(3) 0.1491	(3) 0.6121	(3) 0.3552	(1) 0.4076	(3) 0.2541	(3) 0.5120	(3) 3.1620
MOEA/D-DE-ACA			(1) 0.0957	(1) 0.4968	(1) 0.1647	(3) 0.5191	(2) 0.1711	(1) 0.0502	(1) 3.0132
MOEA/D-DE	DGI	60	(2) 0.0995	(2) 0.5089	(2) 0.1724	(1) 0.4596	(1) 0.1699	(2) 0.0512	(2) 3.1290
NSGA-III			(3) 0.1541	(3) 0.6482	(3) 0.3266	(2) 0.4688	(3) 0.2609	(3) 0.5121	(3) 3.4390
MOEA/D-DE-ACA			(1) 0.0765	(1) 0.5153	(1) 0.1126	(3) 0.5586	(1) 0.1834	(1) 0.0492	(1) 3.3658
MOEA/D-DE	DGI	70	(2) 0.0794	(2) 0.5258	(2) 0.1152	(1) 0.4792	(2) 0.1844	(2) 0.0509	(2) 3.4827
NSGA-III			(3) 0.1185	(3) 0.6923	(3) 0.2262	(2) 0.5146	(3) 0.3186	(3) 0.5034	(3) 3.7559
MOEA/D-DE-ACA			(1) 0.0625	(1) 0.5248	(1) 0.1452	(3) 0.6018	(1) 0.1895	(1) 0.0497	(1) 3.6806
MOEA/D-DE	DGI	80	(2) 0.0698	(2) 0.5491	(2) 0.1699	(1) 0.4926	(2) 0.1895	(2) 0.0514	(2) 3.8031
NSGA-III			(3) 0.1205	(3) 0.7475	(3) 0.2569	(2) 0.5494	(3) 0.3237	(3) 0.5650	(3) 3.9706
MOEA/D-DE-ACA			(1) 0.0748	(1) 0.5625	(1) 0.1395	(3) 0.6226	(1) 0.1791	(1) 0.0442	(1) 4.0533
MOEA/D-DE	DGI	90	(2) 0.0778	(2) 0.5686	(2) 0.1483	(1) 0.5009	(2) 0.1818	(2) 0.0493	(2) 4.1164
NSGA-III			(3) 0.1428	(3) 0.8212	(3) 0.2367	(2) 0.5842	(3) 0.3203	(3) 0.5415	(3) 4.1920
MOEA/D-DE-ACA			(1) 0.0481	(1) 0.5660	(1) 0.1315	(3) 0.6490	(1) 0.2041	(1) 0.0518	(1) 4.3620
MOEA/D-DE	DGI	100	(2) 0.0505	(2) 0.5914	(2) 0.1369	(1) 0.5002	(2) 0.2046	(2) 0.0561	(2) 4.3729
NSGA-III			(3) 0.0999	(3) 0.8708	(3) 0.2325	(2) 0.6097	(3) 0.3062	(3) 0.5912	(3) 4.3839

4.1.1 Test de Friedman para experimentos $t=2$ y $v=17$

Después de evaluar el comportamiento promedio de los tres algoritmos en estos problemas, se ejecutó la prueba estadística no paramétrica de Friedman con el post hoc de Holm similar al experimento anterior. En la **Tabla 11** se puede apreciar que el algoritmo NSGA-III obtiene el ranking 1 en 10, 20 y 30 objetivos mostrando que domina a MOEA/D-DE y a MOEA/D-DE-ACA en 10 y 20 objetivos con 95% de significancia. Además, con 10 objetivos MOEA/D-DE domina a MOEA/D-DE-ACA con el mismo 95% de significancia. En 30 objetivos el ranking de Friedman no es significativo (el valor p obtenido no es menor que 0.05), razón por la cual para estos dos experimentos no se realiza el post hoc de Holm. MOEA/D-DE-ACA obtiene desde 40 hasta 100 objetivos el ranking 1, pero en 50 objetivos este ranking no es estadísticamente significativo.

Basado en Holm, se puede observar que, en 40 objetivos MOEA/D-DE-ACA domina a MOEA/D-DE, con un 95% de significancia. En 60, 90 y 100 objetivos se observa que MOEA/D-DE-ACA y MOEA/D-DE dominan a NSGA-III con 95% de significancia, pero entre MOEA/D-DE-ACA y MOEA/D-DE no se puede establecer ninguna

relación de dominancia. En 70 objetivos se observa que MOEA/D-DE-ACA y MOEA/D-DE dominan a NSGA-III con 95% de significancia y que MOEA/D-DE-ACA domina a MOEA/D-DE con un 90% de significancia. Situación parecida en 80 objetivos, excepto porque la dominancia de MOEA/D-DE-ACA sobre MOEA/D-DE tiene una significancia del 95%.

Tabla 10 Resultados de evaluación en problemas WFG con ACA $v=17$ $t=2$

ALGORITMOS	M	OBJ	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
MOEA/D-DE-ACA			(2) 0.0861	(3) 0.0885	(2) 0.1286	(3) 0.3343	(3) 0.2608	(3) 0.3777	(3) 0.2907	(3) 0.4057	(3) 0.3186
MOEA/D-DE	DGI	10	(1) 0.0614	(2) 0.0787	(3) 0.1452	(2) 0.2937	(2) 0.2365	(2) 0.3279	(2) 0.2625	(2) 0.3584	(2) 0.2874
NSGA-III			(3) 0.1967	(1) 0.0745	(1) 0.1171	(1) 0.1845	(1) 0.1514	(1) 0.1942	(1) 0.1667	(1) 0.1595	(1) 0.1629
MOEA/D-DE-ACA			(2) 3.5750	(1) 0.0901	(2) 0.1941	(3) 0.4078	(2) 0.3499	(3) 0.5497	(3) 0.3604	(3) 0.5203	(2) 0.3877
MOEA/D-DE	DGI	20	(3) 4.9917	(3) 0.1189	(3) 0.2258	(2) 0.3797	(3) 0.3902	(2) 0.4480	(2) 0.3583	(2) 0.4841	(3) 0.4271
NSGA-III			(1) 1.2433	(2) 0.1152	(1) 0.1922	(1) 0.2596	(1) 0.2093	(1) 0.2368	(1) 0.2309	(1) 0.2224	(1) 0.2645
MOEA/D-DE-ACA			(3) 6.6363	(1) 0.0831	(2) 0.1990	(2) 0.4012	(2) 0.4008	(2) 0.5459	(2) 0.4523	(2) 0.5185	(2) 0.4189
MOEA/D-DE	DGI	30	(2) 6.1811	(2) 0.0883	(3) 0.2185	(3) 0.4098	(3) 0.4445	(3) 0.5694	(3) 0.4852	(3) 0.5537	(3) 0.4627
NSGA-III			(1) 1.2260	(3) 0.1048	(1) 0.1941	(1) 0.2495	(1) 0.2300	(1) 0.2123	(1) 0.3180	(1) 0.2400	(1) 0.2804
MOEA/D-DE-ACA			(3) 9.8352	(1) 0.0961	(2) 0.2121	(2) 0.4622	(2) 0.4414	(2) 0.5702	(2) 0.5000	(2) 0.5631	(2) 0.4678
MOEA/D-DE	DGI	40	(2) 8.4161	(2) 0.1071	(3) 0.2346	(3) 0.4944	(3) 0.4781	(3) 0.6023	(3) 0.5410	(3) 0.5903	(3) 0.4954
NSGA-III			(1) 3.1073	(3) 0.1476	(1) 0.1932	(1) 0.3469	(1) 0.3125	(1) 0.3571	(1) 0.3616	(1) 0.3601	(1) 0.3579
MOEA/D-DE-ACA			(3) 7.5191	(1) 0.1306	(2) 0.2220	(1) 0.4881	(2) 0.4437	(1) 0.5710	(2) 0.5151	(2) 0.5669	(2) 0.5131
MOEA/D-DE	DGI	50	(2) 4.8152	(2) 0.1330	(3) 0.2389	(3) 0.5378	(3) 0.4839	(3) 0.6332	(3) 0.5539	(3) 0.6047	(3) 0.5413
NSGA-III			(1) 2.6536	(3) 0.2313	(1) 0.1928	(3) 0.5583	(1) 0.4175	(2) 0.5778	(1) 0.3771	(1) 0.5655	(1) 0.5029
MOEA/D-DE-ACA			(3) 24.898	(1) 0.1106	(3) 0.2454	(1) 0.5102	(2) 0.4871	(1) 0.6280	(2) 0.5645	(1) 0.5975	(2) 0.5750
MOEA/D-DE	DGI	60	(1) 10.071	(2) 0.1393	(2) 0.2310	(2) 0.5305	(3) 0.4976	(2) 0.6399	(3) 0.5729	(2) 0.6029	(1) 0.5691
NSGA-III			(2) 10.842	(3) 0.2291	(1) 0.1887	(3) 0.6509	(1) 0.4845	(3) 0.6495	(1) 0.3919	(3) 0.6341	(3) 0.6724
MOEA/D-DE-ACA			(3) 53.836	(1) 0.1093	(3) 0.2562	(1) 0.5210	(1) 0.5047	(1) 0.6306	(2) 0.5891	(1) 0.5962	(2) 0.5976
MOEA/D-DE	DGI	70	(1) 24.579	(2) 0.1352	(2) 0.2386	(2) 0.5568	(2) 0.5099	(2) 0.6352	(3) 0.5911	(2) 0.6008	(1) 0.5854
NSGA-III			(2) 35.720	(3) 0.2340	(1) 0.1769	(3) 0.7046	(3) 0.5338	(3) 0.6869	(1) 0.4338	(3) 0.6667	(3) 0.7625
MOEA/D-DE-ACA			(2) 18.601	(1) 0.1150	(3) 0.2725	(1) 0.5495	(1) 0.5167	(1) 0.6186	(3) 0.6079	(1) 0.5839	(2) 0.6214
MOEA/D-DE	DGI	80	(1) 14.892	(2) 0.1397	(2) 0.2567	(2) 0.5561	(2) 0.5219	(2) 0.6214	(2) 0.5954	(2) 0.5920	(1) 0.6034
NSGA-III			(3) 20.421	(3) 0.2200	(1) 0.1985	(3) 0.7530	(3) 0.5759	(3) 0.7044	(1) 0.4527	(3) 0.6942	(3) 0.8421
MOEA/D-DE-ACA			(2) 25.382	(1) 0.1391	(3) 0.2795	(1) 0.5549	(1) 0.5384	(1) 0.6244	(3) 0.6219	(1) 0.5938	(2) 0.6537
MOEA/D-DE	DGI	90	(3) 28.030	(2) 0.1841	(2) 0.2651	(2) 0.5850	(2) 0.5469	(2) 0.6339	(2) 0.6203	(3) 0.6026	(1) 0.6257
NSGA-III			(1) 23.195	(3) 0.2300	(1) 0.1833	(3) 0.7908	(3) 0.6172	(3) 0.7418	(1) 0.4832	(3) 0.7242	(3) 0.9550
MOEA/D-DE-ACA			(3) 2.5128	(1) 0.1671	(3) 0.2783	(1) 0.5639	(1) 0.5471	(1) 0.6200	(3) 0.6435	(1) 0.5963	(2) 0.6560
MOEA/D-DE	DGI	100	(2) 1.3551	(2) 0.1988	(2) 0.2674	(2) 0.5894	(2) 0.5589	(2) 0.6324	(2) 0.6318	(2) 0.6170	(1) 0.6305
NSGA-III			(1) 1.2553	(3) 0.2281	(1) 0.1962	(3) 0.8353	(3) 0.6619	(3) 0.7785	(1) 0.5141	(3) 0.7772	(3) 1.0652

Tabla 11 Resultados test de Friedman y post hoc de Holm en DGI para fuerza (t=2) y alfabeto (v=17)

Obj.	MOEA/D-DE-ACA (A)		MOEA/D-DE (B)		NSGA-III (C)	p-value	Significancia	Holm	Tamaño de población	Puntos de referencia																	
10	(3)	2.75	(2)	1.81	(1)	1.44	0.0007	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>○</td><td>○</td></tr> <tr><td>B</td><td>●</td><td>-</td><td>○</td></tr> <tr><td>C</td><td>●</td><td>●</td><td>-</td></tr> </table>		A	B	C	A	-	○	○	B	●	-	○	C	●	●	-	416	12250
	A	B	C																								
A	-	○	○																								
B	●	-	○																								
C	●	●	-																								
20	(2)	2.06	(3)	2.44	(1)	1.50	0.0284	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td></td><td>○</td></tr> <tr><td>B</td><td></td><td>-</td><td>○</td></tr> <tr><td>C</td><td>●</td><td>●</td><td>-</td></tr> </table>		A	B	C	A	-		○	B		-	○	C	●	●	-	514	12334
	A	B	C																								
A	-		○																								
B		-	○																								
C	●	●	-																								
30	(2)	1.88	(3)	2.44	(1)	1.69	0.0874	False	-	679	20467																
40	(1)	1.75	(2)	2.50	(1)	1.75	0.0498	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td></td></tr> <tr><td>B</td><td>○</td><td>-</td><td></td></tr> <tr><td>C</td><td></td><td></td><td>-</td></tr> </table>		A	B	C	A	-	●		B	○	-		C			-	727	22108
	A	B	C																								
A	-	●																									
B	○	-																									
C			-																								
50	(1)	1.63	(3)	2.31	(2)	2.06	0.1441	False	-	753	21675																
60	(1)	1.63	(2)	1.88	(3)	2.50	0.0388	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td></td><td>●</td></tr> <tr><td>B</td><td></td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-		●	B		-	●	C	○	○	-	763	21968
	A	B	C																								
A	-		●																								
B		-	●																								
C	○	○	-																								
70	(1)	1.50	(2)	1.88	(3)	2.63	0.0052	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td></td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B		-	●	C	○	○	-	785	22283
	A	B	C																								
A	-	●	●																								
B		-	●																								
C	○	○	-																								
80	(1)	1.50	(2)	1.81	(3)	2.69	0.0023	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	●	C	○	○	-	807	22988
	A	B	C																								
A	-	●	●																								
B	○	-	●																								
C	○	○	-																								
90	(1)	1.50	(2)	1.94	(3)	2.56	0.0104	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td></td><td>●</td></tr> <tr><td>B</td><td></td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-		●	B		-	●	C	○	○	-	824	23566
	A	B	C																								
A	-		●																								
B		-	●																								
C	○	○	-																								
100	(1)	1.56	(2)	1.88	(3)	2.56	0.0152	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td></td><td>●</td></tr> <tr><td>B</td><td></td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-		●	B		-	●	C	○	○	-	838	23342
	A	B	C																								
A	-		●																								
B		-	●																								
C	○	○	-																								

4.1.2 Tiempo de ejecución para experimentos t=2 y v=17

La **Figura 15** muestra el tiempo promedio de ejecución (Average Execution Time, AET) de los tres algoritmos para todos los datasets DTLZ y WFG con cada número de objetivos, 10 a 100. Se puede apreciar que el tiempo de ejecución de NSGA-III es mucho mayor que el de los otros dos algoritmos, en este caso el crecimiento es cuadrático ($AET = 10635 objectives^2 - 530145 objectives + 9E + 06$ con un $R^2 = 0.9333$). Esto muestra que el procesamiento ejecutado por NSGA-III más allá de las evaluaciones de la función objetivo es mucho mayor al de los otros dos algoritmos.

El tiempo de ejecución de MOEA/D-DE ($AET = 926,52 objectives^2 + 251005 objectives - 1E + 06$ con un $R^2 = 0.9099$) muestra que cuando aumenta la población, el aumento del tiempo es mucho más elevado. Se observa que en 60 objetivos sobrepasa a los otros 2 algoritmos, esto demuestra que usar ACA previamente construido para la ejecución de MOEA/D-DE y liberar este paso de inicialización reduce significativamente el tiempo de procesamiento. Para MOEA/D-DE-ACA el $AET = 753,86 objectives^2 - 56509 objectives + 2E + 06$ con un $R^2 = 0.9093$) que también tiene una tendencia cuadrática donde MOEA/D-DE-ACA es el que presenta el menor tiempo de ejecución (menor constante de multiplicación).

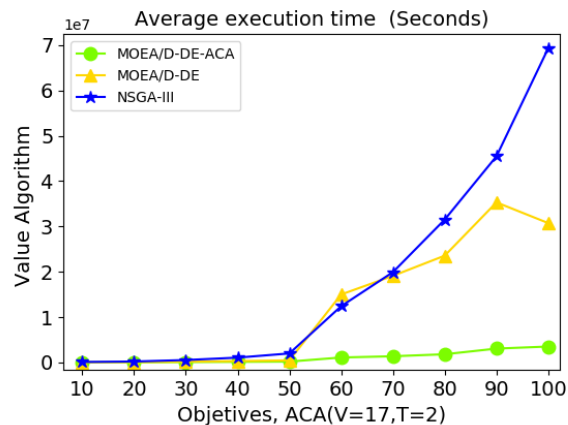


Figura 15 Tiempo promedio de ejecución en segundos para todos los algoritmos en los experimentos con fuerza 2 y alfabeto 17

4.1.3 Resultados por problema para experimentos $t=2$ y $v=17$

La **Figura 16** muestra un gráfico box-plot que resume visualmente la mediana y los cuartiles de los valores DGI obtenidos por los tres algoritmos en los 16 problemas (7 DTLZ y 9 WFG) durante el experimento para $t=2$ y $v=17$ de 10 a 100 objetivos. Se puede observar que, en 11 (DTLZ1 a DTLZ3, DTLZ5 a DTLZ7, WFG2, WFG4 a WFG6 y WFG8 a WFG9) que corresponde al 68.75% de los 16 problemas MOEA/D-DE-ACA obtiene mejores valores de mediana y sus cuartiles están más bajo que los otros algoritmos, seguido en todos los casos por MOEA/D-DE.

En los problemas WFG3, WFG5 y WFG7, NSGA-III obtiene mejores resultados seguidos por MOEA/D-DE-ACA y luego por MOEA/D-DE, en DTLZ4 también NSGA-III obtiene los mejores resultados esta vez seguido por MOEA/D-DE-ACA. En el problema WFG1, MOEA/D-DE obtiene los mejores resultados seguidos por NSGA-III y luego por MOEA/D-DE-ACA, aunque los resultados no son iguales al experimento anterior, si son en general muy similares. Además, si los resultados de las figuras solo incluyeran los valores de 40 a 100 objetivos, que es donde el

algoritmo MOEA/D-DE-ACA obtiene los mejores resultados (conforme a la Tabla 11) el algoritmo propuesto sería el mejor en todos los problemas.

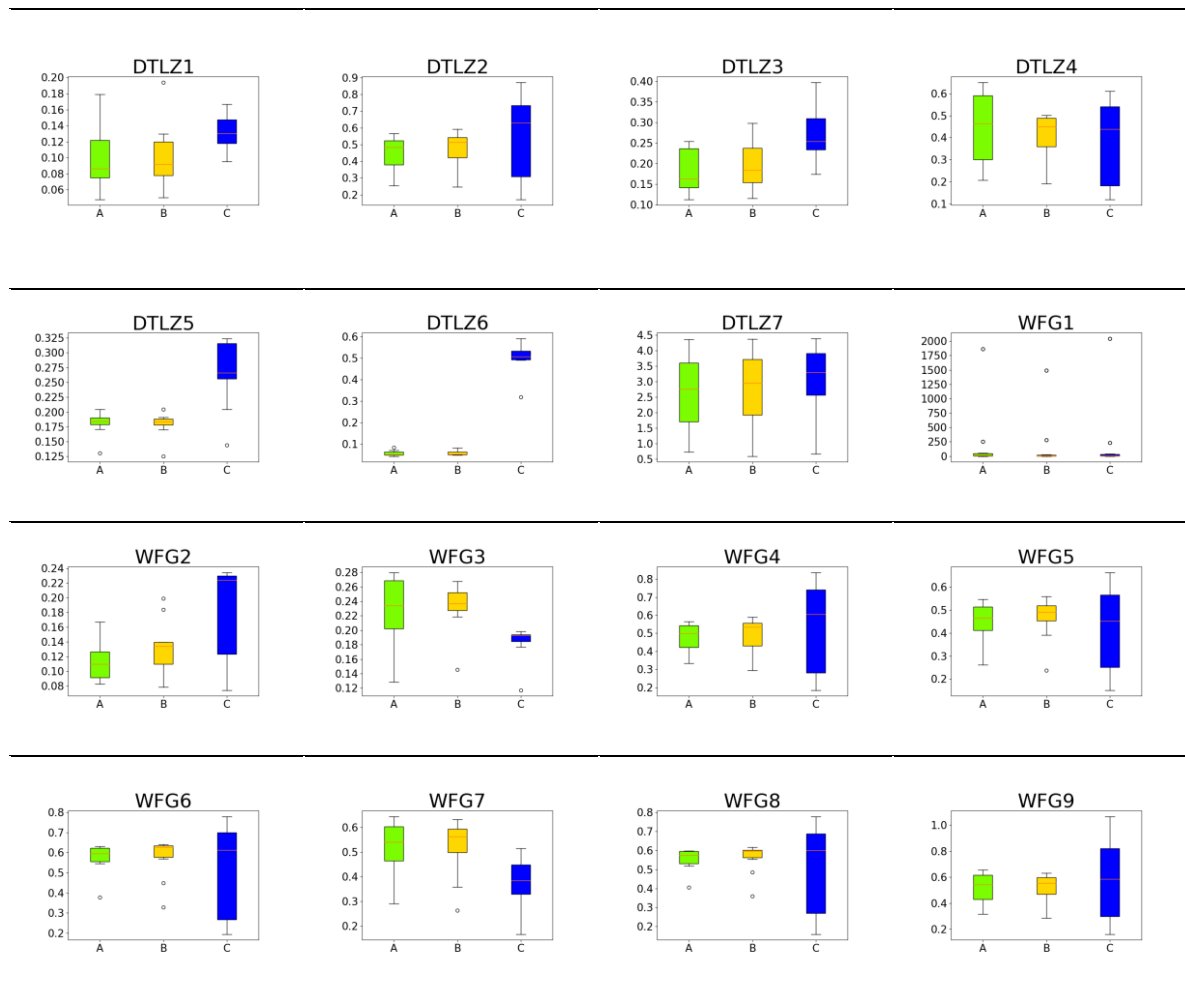


Figura 16 Valores de DGI obtenidos para todos los algoritmos en la evaluación de $v = 17$ y $t = 2$ de 10 a 100 objetivos. A: MOEA / D-DE-ACA, B: MOEA / D-DE y C: NSGA-III

4.2 EXPERIMENTOS CON FUERZA 3 Y ALFABETO 9

En la **Tabla 12** se puede observar lo siguiente:

- 1) MOEA/D-DE-ACA obtiene mejores valores promedio para DGI en problemas DTLZ1 y DTLZ3 con 40 a 100 objetivos.
- 2) En el problema DTLZ7, MOEA/D-DE-ACA obtiene los mejores resultados de DGI desde los 30 hasta los 100 objetivos, mejora el rendimiento ante el número creciente de objetivos para este problema, donde en cada una de las pruebas se observó un nivel alto en funcionamiento para este tipo de problemas con frentes de Pareto complejos.

3) MOEA/D-DE obtiene el mejor DGI para el problema DTLZ6 desde los 10 objetivos y para el problema DTLZ5 desde los 20 objetivos hasta los 100, este algoritmo produce mejores soluciones en la medida en que el número de objetivos aumenta.

4) Finalmente, en el problema DTLZ4, las 2 versiones de MOEA/D son superados en los diferentes objetivos, aunque las diferencias son muy pequeñas.

5) Los algoritmos MOEA/D-DE-ACA y MOEA/D-DE en general lideran los rankings de los valores promedio de DGI desde 20 hasta 100 objetivos.

Tabla 12 Resultados de evaluación en problemas DTLZ con ACA $v=9$ $t=3$

ALGORITMOS	M	OBJ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEA/D-DE-ACA			(3) 0,1296	(2) 0,2233	(2) 0,2236	(3) 0,1661	(3) 0,1572	(2) 0,0944	(3) 0,6029
MOEA/D-DE	DGI	10	(2) 0,1249	(3) 0,2326	(3) 0,2275	(2) 0,1647	(2) 0,1462	(1) 0,0902	(2) 0,5186
NSGA-III			(1) 0,1059	(1) 0,1904	(1) 0,1905	(1) 0,1233	(1) 0,1272	(3) 0,4699	(1) 0,5024
MOEA/D-DE-ACA			(2) 0,1293	(1) 0,2490	(1) 0,2061	(2) 0,1935	(2) 0,1928	(3) 0,0600	(2) 1,0077
MOEA/D-DE	DGI	20	(3) 0,1410	(2) 0,2978	(2) 0,2474	(3) 0,2146	(1) 0,1903	(1) 0,0559	(1) 0,9848
NSGA-III			(1) 0,1138	(3) 0,3374	(3) 0,4183	(1) 0,1874	(3) 0,1935	(2) 0,3295	(3) 1,0118
MOEA/D-DE-ACA			(2) 0,1057	(1) 0,3052	(1) 0,1996	(2) 0,2535	(2) 0,2126	(2) 0,0473	(1) 1,1237
MOEA/D-DE	DGI	30	(3) 0,1225	(3) 0,3442	(2) 0,2269	(3) 0,3056	(1) 0,2065	(1) 0,0440	(2) 1,5969
NSGA-III			(1) 0,0895	(2) 0,3054	(3) 0,2294	(1) 0,2093	(3) 0,2380	(3) 0,3014	(3) 2,5313
MOEA/D-DE-ACA			(1) 0,0754	(2) 0,3735	(1) 0,1482	(2) 0,3605	(2) 0,2545	(2) 0,0489	(1) 1,5200
MOEA/D-DE	DGI	40	(2) 0,0813	(3) 0,3993	(3) 0,1674	(3) 0,3882	(1) 0,2519	(1) 0,0451	(2) 2,1682
NSGA-III			(3) 0,1006	(1) 0,2992	(2) 0,1642	(1) 0,2076	(3) 0,2610	(3) 0,5655	(3) 2,8805
MOEA/D-DE-ACA			(1) 0,0930	(2) 0,3486	(1) 0,1160	(3) 0,4504	(2) 0,2524	(2) 0,0495	(1) 2,0624
MOEA/D-DE	DGI	50	(2) 0,0982	(3) 0,3743	(2) 0,1246	(2) 0,4475	(1) 0,2462	(1) 0,0461	(2) 2,6590
NSGA-III			(3) 0,1579	(1) 0,2799	(3) 0,2056	(1) 0,1925	(3) 0,2853	(3) 0,5293	(3) 3,1753
MOEA/D-DE-ACA			(1) 0,1216	(1) 0,4300	(3) 0,2553	(3) 0,5697	(2) 0,2482	(2) 0,0468	(1) 2,6235
MOEA/D-DE	DGI	60	(2) 0,1245	(3) 0,4534	(2) 0,0848	(2) 0,4849	(1) 0,2427	(1) 0,0447	(2) 3,0310
NSGA-III			(3) 0,1777	(2) 0,4459	(1) 0,1511	(1) 0,3932	(3) 0,2869	(3) 0,5415	(3) 3,5269
MOEA/D-DE-ACA			(1) 0,0952	(1) 0,4804	(1) 0,0801	(3) 0,6287	(2) 0,2508	(2) 0,0514	(1) 3,1768
MOEA/D-DE	DGI	70	(2) 0,1008	(2) 0,5228	(2) 0,1020	(2) 0,5011	(1) 0,2436	(1) 0,0509	(2) 3,4226
NSGA-III			(3) 0,1529	(3) 0,6113	(3) 0,1709	(1) 0,4655	(3) 0,3019	(3) 0,5683	(3) 3,8014
MOEA/D-DE-ACA			(1) 0,0502	(1) 0,5102	(2) 0,0902	(3) 0,6616	(2) 0,2629	(2) 0,0459	(1) 3,5553
MOEA/D-DE	DGI	80	(2) 0,0530	(2) 0,5257	(1) 0,0883	(2) 0,5168	(1) 0,2560	(1) 0,0431	(2) 3,7090
NSGA-III			(3) 0,1188	(3) 0,6422	(3) 0,1637	(1) 0,5088	(3) 0,2893	(3) 0,5593	(3) 4,0793
MOEA/D-DE-ACA			(1) 0,0503	(1) 0,5186	(1) 0,0925	(3) 0,6833	(3) 0,2951	(2) 0,0505	(1) 3,9031
MOEA/D-DE	DGI	90	(2) 0,0563	(2) 0,5389	(2) 0,0932	(1) 0,5285	(1) 0,2890	(1) 0,0486	(2) 4,0176
NSGA-III			(3) 0,1026	(3) 0,6795	(3) 0,1676	(2) 0,5433	(2) 0,2918	(3) 0,4041	(3) 4,3174

En la **Tabla 13** se puede observar lo siguiente:

1) En el problema WFG2 desde los 20 hasta los 90 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI.

2) En los problemas WFG4, WFG6, WFG8 desde los 80 hasta los 90 objetivos, MOEA/D-DE-ACA obtiene mejores valores promedio de DGI, seguido por MOEA/D-DE con diferencias que aumentan poco a poco.

3) En general el algoritmo MOEA/D-DE y MOEA/D-DE-ACA no logran un rendimiento adecuado para este experimento con $v=9$ $t=3$ el algoritmo MOEA/D-DE-

ACA es dominado hasta los 60 objetivos por NSGA-III, se puede observar que el dominio de NSGA-III se caracteriza en problemas WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, WFG9, este algoritmo domina a las 2 versiones de MOEA/D-DE donde desde 10 hasta 60 sus soluciones son mejores en problemas como WFG4 y WFG9 por otra parte NSGA-III domina los problemas WFG3 y WFG7 en este experimento desde los 10 hasta los 90 objetivos.

4) MOEA/D-DE-ACA lidera los experimentos en problemas WFG en cantidades de objetivos grandes 80 y 90 objetivos respectivamente, además la diferencia con NSGA-III no es por pequeñas unidades.

Tabla 13 Resultados de evaluación en problemas WFG con ACA $v=9$ $t=3$

ALGORITMOS	M	OBJ	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
MOEA/D-DE-ACA			(2) 0.1396	(3) 0.0891	(2) 0.1169	(3) 0.3124	(3) 0.2620	(3) 0.4036	(3) 0.2772	(3) 0.4244	(3) 0.3299
MOEA/D-DE	DGI	10	(1) 0.0649	(2) 0.0694	0.1290	(2) 0.2645	(2) 0.2070	(2) 0.3011	(2) 0.2387	(2) 0.3119	(2) 0.2556
NSGA-III			(3) 0.2825	(1) 0.0666	(1) 0.1055	(1) 0.1989	(1) 0.1595	(1) 0.2098	(1) 0.1807	(1) 0.1834	(1) 0.1695
MOEA/D-DE-ACA			(1) 0.1102	(3) 0.0828	(3) 0.1822	(2) 0.3487	(3) 0.3283	(3) 0.5203	(3) 0.3091	(3) 0.5101	(3) 0.3832
MOEA/D-DE	DGI	20	(3) 0.2778	(2) 0.0827	(2) 0.1737	(3) 0.3492	(2) 0.2843	(2) 0.3908	(2) 0.2991	(2) 0.3955	(2) 0.3612
NSGA-III			(2) 0.2713	(1) 0.0812	(1) 0.1499	(1) 0.3142	(1) 0.2197	(1) 0.2896	(1) 0.2776	(1) 0.2664	(1) 0.2587
MOEA/D-DE-ACA			(2) 2.6450	(1) 0.0822	(3) 0.1995	(2) 0.3954	(3) 0.3996	(3) 0.5208	(2) 0.4103	(3) 0.5215	(3) 0.4221
MOEA/D-DE	DGI	30	(3) 6.2116	(2) 0.0959	(2) 0.1817	(3) 0.4025	(2) 0.3952	(2) 0.4552	(3) 0.4110	(2) 0.5012	(2) 0.4123
NSGA-III			(1) 1.3847	(3) 0.1061	(1) 0.1639	(1) 0.3118	(1) 0.2351	(1) 0.2422	(1) 0.3009	(1) 0.2698	(1) 0.2828
MOEA/D-DE-ACA			(2) 16.780	(1) 0.0728	(3) 0.2245	(2) 0.4468	(3) 0.4436	(3) 0.5484	(2) 0.4772	(2) 0.5569	(3) 0.4736
MOEA/D-DE	DGI	40	(3) 25.952	(2) 0.0898	(2) 0.2038	(3) 0.4697	(2) 0.4385	(2) 0.5385	(3) 0.4962	(3) 0.5589	(2) 0.4634
NSGA-III			(1) 9.5468	(3) 0.1199	(1) 0.1734	(1) 0.2930	(1) 0.2358	(1) 0.2266	(1) 0.3582	(1) 0.2490	(1) 0.2892
MOEA/D-DE-ACA			(3) 1.5753	(1) 0.0967	(3) 0.2365	(2) 0.4790	(3) 0.4588	(3) 0.5477	(2) 0.5171	(2) 0.5171	(2) 0.5171
MOEA/D-DE	DGI	50	(2) 0.9944	(2) 0.1082	(2) 0.2115	(3) 0.4797	(2) 0.4567	(2) 0.5723	(3) 0.5363	(3) 0.5363	(3) 0.5363
NSGA-III			(1) 0.5627	(3) 0.1647	(1) 0.1744	(1) 0.2573	(1) 0.2218	(1) 0.2026	(1) 0.3913	(1) 0.3913	(1) 0.3913
MOEA/D-DE-ACA			(3) 2.1443	(1) 0.1159	(3) 0.2553	(2) 0.5046	(3) 0.4733	(2) 0.5917	(2) 0.5438	(2) 0.5909	(3) 0.5327
MOEA/D-DE	DGI	60	(1) 0.5945	(2) 0.1248	(2) 0.2281	(3) 0.5064	(2) 0.4679	(3) 0.6037	(3) 0.5494	(3) 0.6030	(2) 0.5164
NSGA-III			(2) 1.0462	(3) 0.1967	(1) 0.1914	(1) 0.4355	(1) 0.3280	(1) 0.4415	(1) 0.4117	(1) 0.4364	(1) 0.4346
MOEA/D-DE-ACA			(3) 310.73	(1) 0.1165	(3) 0.2586	(1) 0.5343	(3) 0.4810	(2) 0.5997	(2) 0.5794	(2) 0.5923	(2) 0.5616
MOEA/D-DE	DGI	70	(1) 143.45	(2) 0.1229	(2) 0.2308	(2) 0.5565	(2) 0.4793	(3) 0.6266	(3) 0.5872	(3) 0.6138	(1) 0.5484
NSGA-III			(2) 243.12	(3) 0.1993	(1) 0.1783	(3) 0.5898	(1) 0.4112	(1) 0.5865	(1) 0.4510	(1) 0.5649	(3) 0.6234
MOEA/D-DE-ACA			(2) 8.945	(1) 0.1137	(3) 0.2682	(1) 0.5467	(3) 0.4982	(1) 0.5941	(3) 0.6023	(1) 0.5861	(2) 0.5832
MOEA/D-DE	DGI	80	(1) 7.956	(2) 0.1167	(2) 0.2325	(2) 0.5621	(2) 0.4938	(2) 0.6296	(2) 0.5926	(2) 0.6058	(1) 0.5644
NSGA-III			(3) 14.45	(3) 0.1924	(1) 0.1750	(3) 0.6621	(1) 0.4635	(3) 0.6446	(1) 0.4757	(3) 0.6330	(3) 0.7021
MOEA/D-DE-ACA			(1) 1.3280	(1) 0.1033	(3) 0.2723	(1) 0.5717	(2) 0.5139	(1) 0.6090	(3) 0.6199	(1) 0.5847	(1) 0.5790
MOEA/D-DE	DGI	90	(2) 1.7238	(2) 0.1077	(2) 0.2362	(2) 0.5747	(1) 0.5041	(2) 0.6239	(2) 0.5992	(2) 0.5984	(2) 0.5966
NSGA-III			(3) 2.0737	(3) 0.1815	(1) 0.1587	(3) 0.7061	(3) 0.5061	(3) 0.6696	(1) 0.4965	(3) 0.6596	(3) 0.6510

4.2.1 Test de Friedman para experimentos $t=3$ y $v=9$

Después de evaluar el comportamiento promedio de los tres algoritmos en estos problemas, se ejecutó la prueba estadística no paramétrica de Friedman con el post hoc de Holm. Esta prueba se realizó con 2 grados de libertad y los resultados de Holm se evaluaron con un nivel de significancia del 90% y 95%. En la **Tabla 14** se

puede apreciar que el algoritmo MOEA/D-DE-ACA obtiene el ranking número 1 en 90 objetivos, en 20, 30, 40, 60 y 70 objetivos el ranking de Friedman no es estadísticamente significativo (el valor p obtenido no es menor que 0.05), razón por la cual para estos dos experimentos no se realiza el post hoc de Holm. Basado en Holm, se puede observar que, para 10 objetivos el algoritmo NSGA-III domina a MOEA/D-DE-ACA y MOEA/D-DE con un 95% de significancia, para 80 objetivos se determina que MOEA/D-DE y MOEA/D-DE-ACA dominan a NSGA-III con un nivel de significancia del 90% de significancia para finalizar se observa que MOEA/D-DE-ACA y MOEA/D-DE dominan a NSGA-III con un nivel de significancia del 95% respectivamente.

Tabla 14 Resultados test de Friedman y post hoc de Holm en DGI para fuerza ($t=3$) y alfabeto ($v=9$)

Obj.	MOEA/D-DE-ACA (A)	MOEA/D-DE (B)	NSGA-III (C)	p-value	Significancia	Holm	Tamaño de población	Puntos de referencia																			
10	(3)	2.69	(2)	2.06	(1)	1.25	0.000245	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>○</td><td>○</td></tr> <tr><td>B</td><td>●</td><td>-</td><td>○</td></tr> <tr><td>C</td><td>●</td><td>●</td><td>-</td></tr> </table>		A	B	C	A	-	○	○	B	●	-	○	C	●	●	-	729	18806
	A	B	C																								
A	-	○	○																								
B	●	-	○																								
C	●	●	-																								
20	(3)	2.25	(2)	2.06	(1)	1.69	0.269146	False	-	1377	28906																
30	(2)	2.06	(3)	2.25	(1)	1.69	0.269146	False	-	1449	38232																
40	(2)	2.00	(3)	2.31	(1)	1.69	0.209611	False	-	1449	38673																
50	(2)	2.00	(3)	2.25	(1)	1.75	0.367879	False	-	1449	38720																
60	(3)	2.13	(2)	2.06	(1)	1.81	0.645648	False	-	1449	39889																
70	(1)	1.88	(2)	1.94	(3)	2.19	0.645648	False	-	1449	39667																
80	(2)	1.81	(1)	1.69	(3)	2.50	0.046770	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td></td><td>●</td></tr> <tr><td>B</td><td></td><td>-</td><td>●</td></tr> <tr><td>C</td><td></td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-		●	B		-	●	C		○	-	1457	38891
	A	B	C																								
A	-		●																								
B		-	●																								
C		○	-																								
90	(1)	1.68	(2)	1.75	(3)	2.56	0.022092	True	<table border="1"> <tr><td></td><td>A</td><td>B</td><td>C</td></tr> <tr><td>A</td><td>-</td><td>●</td><td>●</td></tr> <tr><td>B</td><td>○</td><td>-</td><td>●</td></tr> <tr><td>C</td><td>○</td><td>○</td><td>-</td></tr> </table>		A	B	C	A	-	●	●	B	○	-	●	C	○	○	-	1457	39190
	A	B	C																								
A	-	●	●																								
B	○	-	●																								
C	○	○	-																								

4.2.2 Tiempo de ejecución para experimentos $t=3$ y $v=9$

La $\text{Error! No se encuentra el origen de la referencia.}$ muestra el tiempo promedio de ejecución (Average Execution Time, AET) de los tres algoritmos para todos los datasets DTLZ y WFG con cada número de objetivos, 10 a 90. Se puede apreciar que el tiempo de ejecución de NSGA-III disminuye a comparación de los anteriores experimentos, en este caso el crecimiento es cuadrático ($AET = 1342 \text{ objectives}^2 -$

4690 *objectives* + 206500 con un $R^2 = 0.9795$). En esta ejecución se observa que el procesamiento ejecutado por MOEA/D-DE es mucho mayor a los otros dos algoritmos, ya que todos ejecutan el mismo número de EFOs, Por otra parte MOEA/D-DE-ACA ($AET = 87,379 \text{ objectives}^2 - 1263,1 \text{ objectives} + 48970$ con un $R^2 = 0.8843$) al igual que en los experimentos previos, presenta el menor tiempo de ejecución (menor constante de multiplicación).

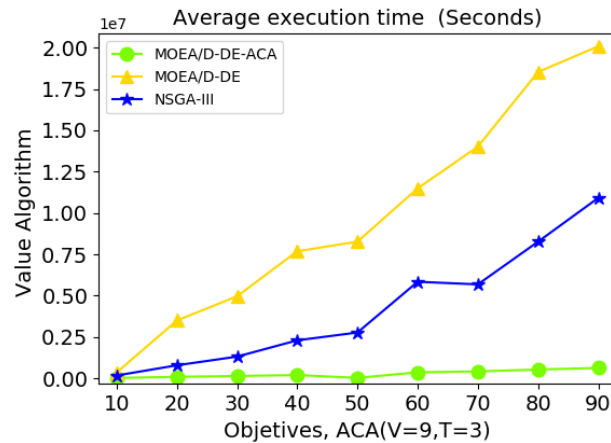


Figura 17 Tiempo promedio de ejecución en segundos para todos los algoritmos en los experimentos con fuerza 3 y alfabeto 9

El tiempo de ejecución de MOEA/D-DE también es cuadrático ($AET = 901,2 \text{ objectives}^2 + 153215 \text{ objectives} - 639727$ con un $R^2 = 0.9881$) donde el enfoque de MOEA/D-DE demuestra que cada vez que se aumenta el número de vectores de peso aumenta en demasía el tiempo de procesamiento, desde 20 hasta 90 objetivos.

Después de realizar los tres experimentos y analizar el tiempo de ejecución en cada uno de los algoritmos se identifica que la incorporación de ACAs, independientemente de la fuerza y el alfabeto en MOEA/D-DE mejora los resultados. Utilizando ACA $v=9, t=2$ la propuesta es 57.7% más ágil en comparación con NSGA-III y 2.7% mejor que MOEA/D-DE, de igual manera utilizando ACA $v=17, t=2$ MOEA/D-DE-ACA es 22% mejor que NSGA-III y 7% mejor que MOEA/D-DE, finalmente utilizando ACA $v=9, t=3$ se observa que MOEA/D-DE-ACA es un 15.42% mejor que MOEA/D-DE y 10.3% mejor que NSGA-III. Se identifica que la generación aleatoria de vectores de pesos en MOEA/D-DE cuando la población es grande (824-1457) el algoritmo aumenta mucho más el tiempo de ejecución ante el número creciente de objetivos, lo cual demuestra que la incorporación del enfoque propuesto de usar ACAs mejora significativamente el rendimiento de MOEA/D-DE en cada caso de prueba evaluado.

4.2.3 Resultados por problema para experimentos $t=3$ y $v=9$

La **Figura 18** muestra un gráfico box-plot con los resultados de los tres algoritmos en la ejecución para $t=3$, $v=9$ donde se visualiza fácilmente la mediana y los cuartiles de los valores DGI obtenidos por los tres algoritmos en los 16 problemas evaluados.

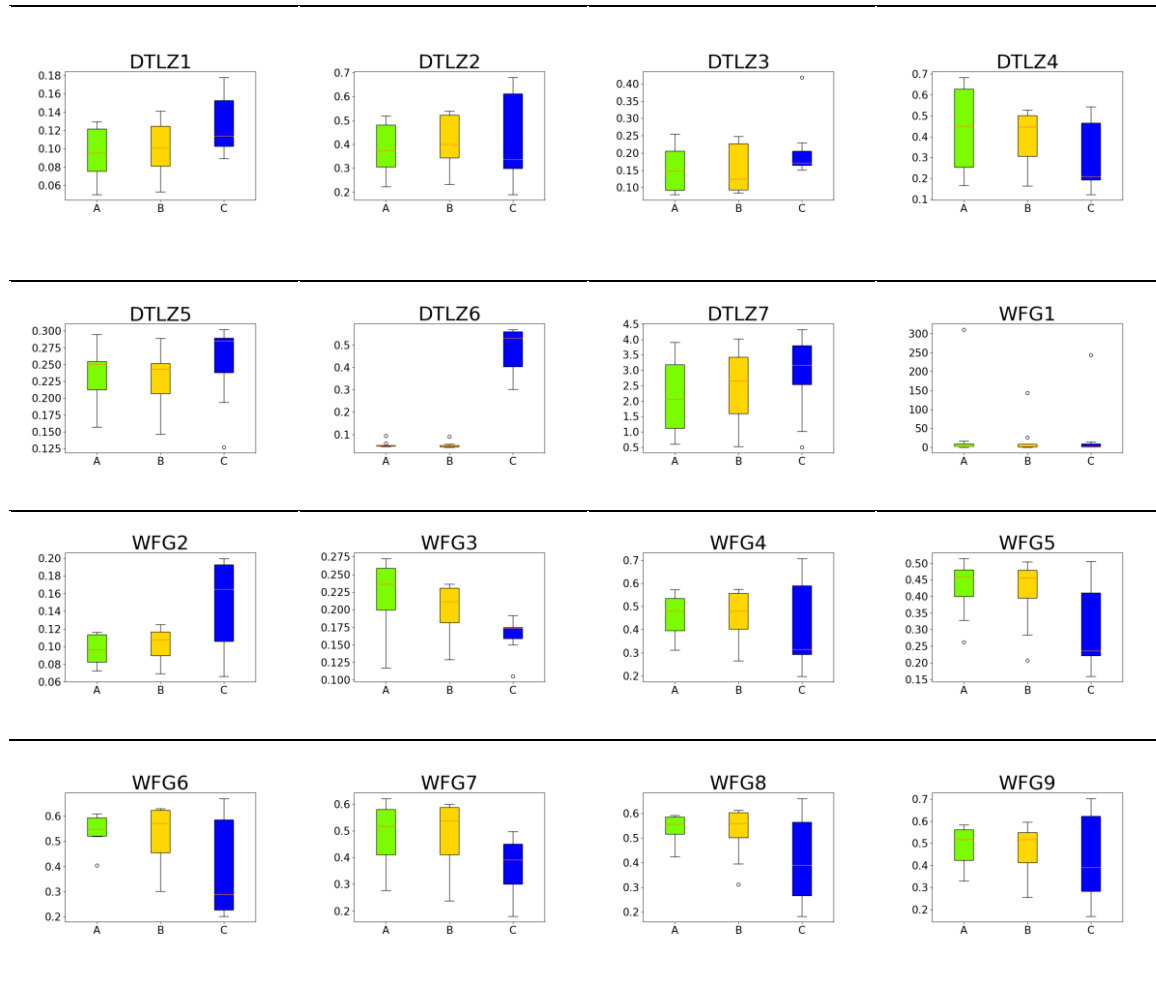


Figura 18 Valores de DGI obtenidos para todos los algoritmos en la evaluación de $v = 17$ y $t = 2$ de 10 a 100 objetivos. A: MOEA / D-DE-ACA, B: MOEA / D-DE y C: NSGA-III

Se puede observar que, MOEA/D-DE-ACA obtiene mejores (menores) valores de DGI y su promedio es el menor en problemas DTLZ como DTLZ1, DTLZ2, DTLZ6, DTLZ7 y en problemas WFG en problemas tales como WFG1 y WFG2, por otra parte, el rendimiento de NSGA-III en todos los problemas mejora, a diferencia del experimento anterior, donde en 9 de los 16 problemas evaluados supera levemente a los resultados obtenidos por MOEA/D-DE-ACA y MOEA/D-DE.

CAPÍTULO 5

5 CONCLUSIONES Y TRABAJO FUTURO

Este trabajo propone la definición de vectores de peso en MOEA/D-DE basado en Augmented Covering Arrays (ACAs), en una nueva versión del algoritmo denominado MOEA/D-DE-ACA. Esta nueva versión se comparó con la versión original de MOEA/D-DE y NSGA-III en 7 problemas DTLZ y 9 problemas WFG de 10 a 100 objetivos usando poblaciones pequeñas (alfabeto $v=9$ y fuerza $t=2$), medianas ($v=17$, $t=2$) y grandes ($v=9$, $t=3$).

Los resultados de la evaluación de los tres algoritmos en los diferentes (16) MaOPS muestra que todavía hay muchas cosas por hacer para definir un algoritmo que obtenga los mejores resultados del estado del arte sin importar las características del Frente de Pareto de un problema y el número de objetivos de este.

En relación con la hipótesis inicialmente planteada en esta investigación se puede concluir que con un valor bajo de fuerza $t=2$ y un valor bajo de alfabeto $v = 9$ o $v = 17$ (poblaciones pequeñas con 136 a 288 soluciones y medianas con 416 a 838 soluciones) el algoritmo MOEA/D-DE-ACA obtiene mejores resultados de DGI que MOEA/D-DE y que NSGA-III entre 40 y 100 objetivos sin importar las características de los problemas y las formas de sus frentes de Pareto, lo que implica que la iniciación de los pesos es más adecuada con los ACA y que el tiempo de ejecución se reduce significativamente, a un 40.7% y un 8.9% del tiempo ejecutado por MOEA/D-DE y NSGA-III. Cuando se usa una fuerza $t = 3$ (población grande con 729 a 1457 soluciones), los resultados son similares entre MOEA/D-DE-ACA y MOEA/D-DE, es decir, no hay una diferencia estadísticamente significativa entre MOEA/D-DE-ACA y MOEA/D-DE, excepto en 90 objetivos donde MOEA/D-DE-ACA domina, pero el tiempo de ejecución con estas poblaciones se reduce mucho más, a sólo un 2.7% del tiempo de ejecución de MOEA/D-DE, es decir, ahorra un 97.3% del tiempo de ejecución.

En los experimentos con poblaciones pequeñas y medianas, los dos algoritmos, MOEA/D-DE-ACA y MOEA/D-DE, dominan a NSGA-III, pero cuando se tienen 10 o 20 objetivos y poblaciones medianas los resultados son mejores con NSGA-III. Con una población grande, NSGA-III obtiene mejores resultados que los otros dos algoritmos de 10 a 70 objetivos, pero esos resultados no dominan estadísticamente a los obtenidos por MOEA/D-DE-ACA y por MOEA/D-DE. MOEA/D-DE-ACA en todos los casos se ejecuta más rápidamente que NSGA-III. En poblaciones pequeñas solo el 4.8% del tiempo usado por NSGA-III, en poblaciones medianas solo el 7% del usado por NSGA-III y en poblaciones grandes, sólo el 6.2% del usado por NSGA-III.

Como trabajo futuro, el grupo de investigación espera realizar la comparación directa del esquema de inicialización propuesto frente a otros esquemas de inicialización como el uso del método simplex, Diseño uniforme, Monte Carlo, método Simplex-Lattice. Además, usar el esquema de inicialización propuesto en versiones más reciente de MOEA/D como en adaptive MOEA/D [64], una versión que adapta la dirección de los vectores de peso y mejora los operadores de selección.

Por otra parte, se espera implementar CAs y ACAs en el algoritmo NSGA-III para la definición de los puntos de referencia y así realizar la respectiva evaluación con la versión actual del estado del arte, determinando si los CAs o los ACAs aportan a una mejor definición de los puntos de referencia del algoritmo.

Finalmente, se espera definir una nueva estructura de CA donde en cada fila se cuente con un mayor número de ceros, de esta forma se logra evaluar mejor la interacción de conjuntos (vectores de peso) que tienen un menor número de objetivos. Eso sí, que esta nueva estructura mantenga un amplio muestreo de las LDEU basado en el parámetro de fuerza t y alfabeto v .

CAPÍTULO 6

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. V. Marroqu and C. Cimat-monterrey, "Optimización Multi- - -- objetivo Contenido," *Conacyt*, vol. 01, no. 2015-02-14, pp. 02-24, 2015.
- [2] S. Chand and M. Wagner, "Evolutionary many-objective optimization: A quick-start guide," *Surv. Oper. Res. Manag. Sci.*, vol. 20, no. 2, pp. 35–42, 2015.
- [3] S. Jiang and S. Yang, "An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *TIK-Report*, vol. 103, 2001.
- [6] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
- [7] E. J. Hughes, "Multiple single objective Pareto sampling," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, 2003, vol. 4, p. 2678–2684 Vol.4.
- [8] Y. Hui, Y. Xin, and S. Min, "Particle swarm optimization route planner algorithm for air vehicle," in *2010 International Conference on Computer and Information Application*, 2010, pp. 319–322.
- [9] Y. Li, X. Bai, and Z. Wu, "The determination of optimal design plan of the Sha-He aqueduct," in *2010 2nd IEEE International Conference on Information Management and Engineering*, 2010, pp. 319–323.
- [10] E. Ruano, C. Cobos, and J. Torres-Jimenez, "Transit Network Frequencies-Setting Problem Solved Using a New Multi-Objective Global-Best Harmony Search Algorithm and Discrete Event Simulation," in *Advances in Soft Computing: 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Canc{ú}n, Mexico, October 23--28, 2016, Proceedings, Part II*, O. Pichardo-Lagunas and S. Miranda-Jiménez, Eds. Cham: Springer International Publishing, 2017, pp. 341–352.
- [11] B. Li, J. Li, K. Tang, and X. Yao, "Many-Objective Evolutionary Algorithms: A Survey," *ACM Comput. Surv.*, vol. 48, no. 1, p. 13:1--13:35, 2015.

- [12] H. Li and Q. Zhang, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.
- [13] W. Zheng, Y. Tan, X. Fang, and S. Li, "An Improved MOEA/D with Optimal DE Schemes for Many-Objective Optimization Problems," *Algorithms*, vol. 10, no. 3, 2017.
- [14] B. Li, J. Li, K. Tang, and X. Yao, "An improved Two Archive Algorithm for Many-Objective optimization," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 2869–2876.
- [15] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.
- [16] S. S. Bhagavatula, S. G. Sanjeevi, D. Kumar, and C. K. Yadav, "Multi-objective indicator based evolutionary algorithm for portfolio optimization," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014, pp. 1206–1210.
- [17] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "An Interactive Simple Indicator-Based Evolutionary Algorithm (I-SIBEA) for Multiobjective Optimization Problems," in *Evolutionary Multi-Criterion Optimization*, 2015, pp. 277–291.
- [18] J. Luo, Q. Liu, Y. Yang, X. Li, M. Chen, and W. Cao, "An artificial bee colony algorithm for multi-objective optimisation," *Appl. Soft Comput.*, vol. 50, pp. 235–251, 2017.
- [19] H. Xie, J. Li, and H. Xue, "A survey of dimensionality reduction techniques based on random projection," *CoRR*, vol. abs/1706.04371, 2017.
- [20] H. Aguirre and K. Tanaka, "Adaptive ϵ -ranking on MNK-Landscapes," in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*, 2009, pp. 104–111.
- [21] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between Neighborhood Size and MOEA/D Performance on Many-Objective Problems," in *Evolutionary Multi-Criterion Optimization*, 2013, pp. 459–474.
- [22] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [23] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.
- [24] X. Cai, Z. Mei, and Z. Fan, "A Decomposition-Based Many-Objective Evolutionary Algorithm With Two Types of Adjustments for Direction Vectors," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–14, 2017.
- [25] J. Torres-jimenez and A. Rodriguez-cristerna, "CAAI Transactions on

- Intelligence Technology Metaheuristic post-optimization of the NIST repository of covering arrays,” *CAAI Trans. Intell. Technol.*, pp. 6–13, 2017.
- [26] G. Tzanakis, L. Moura, D. Panario, and B. Stevens, “Constructing new covering arrays from LFSR sequences over finite fields,” *Discrete Math.*, vol. 339, no. 3, pp. 1158–1171, 2016.
- [27] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2419–2426.
- [28] S. Bechikh, M. Elarbi, and L. Ben Said, “Many-objective Optimization Using Evolutionary Algorithms: A Survey,” in *Recent Advances in Evolutionary Multi-objective Optimization*, S. Bechikh, R. Datta, and A. Gupta, Eds. Cham: Springer International Publishing, 2017, pp. 105–137.
- [29] N. de la P. J. Lopez Buenos Aires, Argentina, Universidad, *Optimización multiobjetivo: aplicaciones a problemas del mundo real*. 2013.
- [30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable Test Problems for Evolutionary Multiobjective Optimization,” in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. London: Springer London, 2005, pp. 105–145.
- [31] S. Huband, L. Barone, L. While, and P. Hingston, “A Scalable Multi-objective Test Problem Toolkit,” in *Evolutionary Multi-Criterion Optimization*, 2005, pp. 280–295.
- [32] S. Mirjalili and A. Lewis, “Novel performance metrics for robust multi-objective optimization algorithms,” *Swarm Evol. Comput.*, pp. 1–23, 2014.
- [33] J. J. Durillo and A. J. Nebro, “JMetal: A Java framework for multi-objective optimization,” *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
- [34] A. Trivedi, D. Srinivasan, and S. Member, “A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition,” vol. 21, no. 3, pp. 440–462, 2017.
- [35] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, “MOEA/D with Adaptive Weight Adjustment,” *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [36] X. Cai, Z. Mei, and Z. Fan, “A Decomposition-Based Many-Objective Evolutionary Algorithm With Two Types of Adjustments for Direction Vectors,” *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–14, 2018.
- [37] F. Gu and Y. Cheung, “Self-Organizing Map-Based Weight Design for Decomposition-Based Many-Objective Evolutionary Algorithm,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211–225, 2018.
- [38] H. Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective SubproblemsLiu, F. Gu, and Q. Zhang, “Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp.

- 450–455, 2014.
- [39] C. D. y X. Lei, “Un algoritmo evolutivo multiobjetivo basado en descomposición con ajuste de peso adaptativo,” *Complexity*. pp. 1–20, 2018.
- [40] J. Qiao, H. Zhou, C. Yang, and S. Yang, “A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty,” *Appl. Soft Comput.*, vol. 74, pp. 190–205, 2019.
- [41] H. Xu, W. Zeng, D. Zhang, and X. Zeng, “MOEA/HD: A Multiobjective Evolutionary Algorithm Based on Hierarchical Decomposition,” *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.
- [42] C. von Lüken, B. Barán, and C. Brizuela, “A survey on multi-objective evolutionary algorithms for many-objective problems,” *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 707–756, Jul. 2014.
- [43] B. Li, K. Tang, J. Li, and X. Yao, “Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924–938, 2016.
- [44] X. Zhang, Y. Tian, and Y. Jin, “A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, 2015.
- [45] K. Deb and H. Jain, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.
- [46] C. Alberto and C. Lozada, *Algoritmos Metaheurísticos para Optimización de Muchos Objetivos*, Universida. Popayán-Cauca-Colombia, 2019.
- [47] A. Hartman and L. Raskin, “Problems and algorithms for covering arrays,” *Discrete Math.*, vol. 284, no. 1, pp. 149–156, 2004.
- [48] J. Timaná, C. A. Cobos-lozada, and J. Torres-jimenez, “Metaheuristic algorithms for building Covering Arrays : A review Algoritmos metaheurísticos para construir Covering Arrays : Revisión Algoritmos metaheurísticos para construir Covering Arrays :,” *Rev. Fac. Ing.*, vol. 25, no. 43, pp. 31–45, 2016.
- [49] R. Kuhn, Y. Lei, and R. Kacker, “Practical Combinatorial Testing: Beyond Pairwise,” *IT Prof.*, vol. 10, no. 3, pp. 19–23, May 2008.
- [50] H. Ordoñez, J. Torres-Jimenez, A. Ordoñez, and C. Cobos, “Clustering Business Process Models Based on Multimodal Search and Covering Arrays,” in *Advances in Soft Computing*, 2017, pp. 317–328.
- [51] E. Ruano-Daza, C. Cobos, J. Torres-Jimenez, M. Mendoza, and A. Paz, “A multiobjective bilevel approach based on global-best harmony search for defining optimal routes and frequencies for bus rapid transit systems,” *Appl. Soft Comput.*, vol. 67, pp. 567–583, 2018.
- [52] H. Sato, “Inverted PBI in MOEA/D and Its Impact on the Search Performance

- on Multi and Many-objective Optimization,” in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 645–652.
- [53] D. S. Johnson, “Approximation algorithms for combinatorial problems,” *J. Comput. Syst. Sci.*, vol. 9, no. 3, pp. 256–278, 1974.
- [54] L. Lovász, “On the ratio of optimal integral and fractional covers,” *Discrete Math.*, vol. 13, no. 4, pp. 383–390, 1975.
- [55] S. K. Stein, “Two combinatorial covering theorems,” *J. Comb. Theory, Ser. A*, vol. 16, no. 3, pp. 391–397, 1974.
- [56] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
- [57] C. Zhou, G. Dai, C. Zhang, X. Li, and K. Ma, “Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems,” *Inf. Sci. (Ny)*, vol. 465, pp. 232–247, Oct. 2018.
- [58] B. Khan, S. Hanoun, M. Johnstone, C. P. Lim, D. Creighton, and S. Nahavandi, “A Scalarization-based Dominance Evolutionary Algorithm for Many-objective Optimization,” *Inf. Sci. (Ny)*, Sep. 2018.
- [59] J. Zou, Y. Zhang, S. Yang, Y. Liu, and J. Zheng, “Adaptive neighborhood selection for many-objective optimization problems,” *Appl. Soft Comput.*, vol. 64, pp. 186–198, 2018.
- [60] Q. Lin, Q. Zhu, P. Huang, J. Chen, Z. Ming, and J. Yu, “A novel hybrid multi-objective immune algorithm with adaptive differential evolution,” *Comput. Oper. Res.*, vol. 62, pp. 95–111, Oct. 2015.
- [61] R. Sengupta and S. Saha, “Reference point based archived many objective simulated annealing,” *Inf. Sci. (Ny)*, vol. 467, pp. 725–749, Oct. 2018.
- [62] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.
- [63] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [64] J. Zhou *et al.*, “A Decomposition based Evolutionary Algorithm with Direction Vector Adaption and Selection Enhancement,” *Inf. Sci. (Ny)*, 2019.