

**UNIVERSIDAD AUTÓNOMA DE SINALOA DOCTORADO
EN CIENCIAS DE LA INFORMACIÓN**



**“SOLUCIÓN AL PROBLEMA DE CLASIFICACIÓN
ORDINAL MULTICRITERIO CON RESTRICCIONES
AL TAMAÑO DE LAS CATEGORÍAS UTILIZANDO
RELACIONES DE NO-INFERIORIDAD Y
ALGORITMOS EVOLUTIVOS MULTI OBJETIVO”**

TESIS

como requisito para obtener el grado de

DOCTOR EN CIENCIAS DE LA INFORMACIÓN

Presentada Por:

Omar Ivan Gaxiola Sánchez

Directores de tesis:

Dr. Eduardo René Fernández González

Dr. Jorge Adalberto Navarro Castillo

Culiacán, Sinaloa a 13 de febrero de 2019.

Agradecimientos

La concepción, desarrollo y conclusión de este trabajo de tesis, no hubiese sido posible sin el apoyo de una serie de personas e instituciones. Mi agradecimiento a todos los que formaron parte de este proyecto.

Mi reconocimiento y gratitud a mis directores de tesis, Dr. Eduardo René Fernández González y Dr. Jorge Adalberto Navarro Castillo, por su paciencia, enseñanza, dedicación y recomendaciones académicas y personales durante el desarrollo de este trabajo.

Agradezco al Tecnológico Nacional de México y al Instituto Tecnológico de Culiacán por el apoyo académico, y laboral, para dedicarme de manera exclusiva a este doctorado. Al Consejo Nacional de Ciencia y Tecnología y al Programa de Mejoramiento del Profesorado por su soporte económico.

Agradezco a la Universidad Autónoma de Sinaloa y a la Facultad de Informática por adoptarme como estudiante del programa de Doctorado en Ciencias de la Información.

A mis amigos y compañeros del programa de doctorado y maestría, Universidad Autónoma de Sinaloa, Facultad de Informática Culiacán UAS. Gracias a todos ustedes.

A mis padres por darme la vida, amor, cariño y consejos. Por apoyarme y motivarme siempre en los proyectos que emprendo.

A mis hermanos y toda mi familia por su apoyo incondicional.

Un especial agradecimiento para mi esposa Julieta y a nuestros hijos Isaac, Izan e Irvin. Muchos fueron los momentos que no estuve con ellos, sin embargo, estuvieron siempre apoyándome y motivándome para continuar con este compromiso.

Gracias.

Siempre CJOI.

Índice

Lista de tablas	i
Lista de figuras.....	ii
Lista de símbolos y acrónimos.....	iii
Resumen.....	v
Abstract.....	vi
1. Introducción.....	1
1.1. Generalidades.....	1
1.1.1. Enfoques principales del análisis de la decisión.....	4
1.2. Antecedentes.....	9
1.3. Descripción del problema.....	13
1.4. Preguntas de investigación	16
1.5. Alcance y limitaciones del trabajo de investigación.....	16
1.6. Objetivos.....	17
1.6.1. Objetivo General.....	17
1.6.2. Objetivos Específicos.....	17
1.7. Estructura de la tesis.....	17
2. Fundamentos teóricos.....	19
2.1. Definiciones y conceptos básicos.....	19
2.2. Tipo de problema a tratar.....	20
2.3. Problemas continuos.....	20
2.3.1. Optimización de un objetivo.....	21
2.3.2. Optimización multiobjetivo.....	22
2.4. Problemas discretos.....	25
2.4.1. Clasificación con restricción en el tamaño de las categorías.....	26
2.4.2. Definición de un problema CSP.....	29
2.5. ELECTRE TRI-B.....	30
2.5.1. Construcción de la relación de no-inferioridad S.....	31

2.5.2. Procedimiento de construcción.....	31
2.5.3. Procedimiento de explotación.....	33
2.6. Análisis de Desagregación de Preferencias (PDA).....	35
2.7. Metaheurísticas.....	37
2.7.1. Algoritmo genético.....	39
2.7.1.1. Algoritmo.....	40
2.7.1.2. Selección.....	42
2.7.1.2.1. Selección por ruleta.....	42
2.7.1.2.2. Selección por torneo.....	44
2.7.1.3. Cruce.....	45
2.7.1.4. Mutación.....	46
2.7.1.5. Elitismo.....	46
2.7.1.6. Algoritmos genéticos vs otras técnicas evolutivas.....	47
2.7.1.7. Aplicaciones.....	47
2.7.2. Algoritmo Evolutivo diferencial.....	47
2.7.2.1. Algoritmo.....	48
2.7.2.2. Aplicaciones.....	50
2.7.3. Algoritmo Evolutivo Multiobjetivo basado en Descomposición (MOEA/D).....	50
2.7.3.1. Algoritmo.....	51
3. Implementación del Análisis de Desagregación de Preferencias PDA...	54
3.1. Parámetros a extraer con el método PDA.....	54
3.2. Implementación del método PDA mediante un algoritmo genético....	55
3.2.1. Generación de una población semilla.....	60
3.2.2. Experimentación del algoritmo genético con diferente técnica de selección y reemplazo.....	61
3.3. Evolución diferencial.....	65
3.3.1. Esquema de transformación.....	66
3.3.2. Implementación del método PDA mediante un algoritmo evolutivo diferencial.....	69

3.3.3. Generación de una población semilla.....	72
3.3.4. Experimentación.....	73
3.4 Comparación de resultados de los dos métodos PDA	75
4. Enfoque multiobjetivo propuesto para el problema de clasificación ordinal multicriterio con restricciones sobre los tamaños de las categorías.....	77
4.1. Formulación del problema CSP como un problema de optimización multiobjetivo.....	78
4.2. Implementación del algoritmo multiobjetivo MOEA/D.....	79
4.2.1. Implementación del algoritmo multiobjetivo MOEA/D sin utilizar el método PDA.....	81
4.2.2. Implementación del algoritmo MOEA/D utilizando el método PDA.....	82
4.2.2.1. Utilización de la población semilla.....	82
4.3. Generador de instancias.....	84
4.4. Experimentación.....	85
4.5. Resultados.....	86
4.5.1. Sin utilizar el método PDA.....	87
4.5.2. Utilizando el método PDA.....	89
4.6. Comparación de resultados	92
4.7. Ordenamiento de los objetos al interior de las categorías.....	93
5. Conclusiones.....	95
5.1. Análisis del cumplimiento de objetivos y de las respuestas a las preguntas de investigación.....	95
5.2. Conclusiones.....	99
5.3. Trabajo futuro.....	100
Anexo 1. Conjunto de entrenamiento.....	102
Anexo 2. Resultados de la experimentación sin utilizar el método PDA.....	104

Anexo 3. Resultados de la experimentación utilizando el método PDA.....	120
Anexo 4. Prueba no paramétrica de Mann-Whitney a los resultados del método PDA implementado con un algoritmo genético y con un algoritmo evolutivo diferencial.....	136
Anexo 5. Prueba no paramétrica de Mann-Whitney a los resultados del método PDA implementado con un algoritmo genético y con un algoritmo evolutivo diferencial.....	152
Referencias	160

Lista de tablas

Número	Título	Pág.
Tabla 1.1.	Ejemplo de alternativas no dominadas.....	2
Tabla 1.2.	Publicaciones asociadas a los problemas CSP	11
Tabla 3.1.	Ejemplos de longitud, puntos de cruce y puntos de mutación para individuos	59
Tabla 3.2.	Resultados de los experimentos para las tres configuraciones implementadas.....	63
Tabla 3.3.	Resultados de los experimentos con la misma población inicial.....	64
Tabla 3.4.	Comparativa de los resultados de los experimentos.....	75
Tabla 4.1.	Valores utilizados en la experimentación.....	86
Tabla 4.2.	Efectividad por categoría sin método PDA.....	87
Tabla 4.3.	Efectividad por criterio sin método PDA.....	87
Tabla 4.4.	Efectividad por tamaño del conjunto A sin método PDA.....	88
Tabla 4.5.	Efectividad por número de objetos en cada categoría del conjunto T sin método PDA.....	88
Tabla 4.6.	Tiempos de ejecución sin método PDA.....	89
Tabla 4.7.	Efectividad por categoría con método PDA.....	89
Tabla 4.8.	Efectividad por criterio con método PDA.....	90
Tabla 4.9.	Efectividad por tamaño del conjunto A con método PDA.....	90
Tabla 4.10.	Efectividad por número de objetos en cada categoría del conjunto T con método PDA.....	91
Tabla 4.11.	Tiempos de ejecución con método PDA.....	91

Lista de figuras

Número	Título	Pág.
Figura 1.1.	Problemas de la toma de decisiones.....	3
Figura 1.2.	Problemática planteada.....	14
Figura 1.3.	Propuesta de solución al problema	15
Figura 2.1.	Ejemplo de problemas discretos y continuos.....	20
Figura 2.2.	Frontera de Pareto para Max $T(x)$ y Max	24
Figura 2.3.	Definición de las categorías usando perfiles como límites.....	30
Figura 2.4.	Paradigmas de agregación y desagregación en MCDA.....	35
Figura 2.5.	Ejemplo de la codificación (mediante cadenas binarias) usada tradicionalmente con los algoritmos genéticos.....	41
Figura 2.6.	Cruce de un punto aleatorio.....	45
Figura 2.7.	Operador de cruce exponencial del algoritmo DE.....	49
Figura 3.1.	Método de análisis de desagregación de preferencias.....	54
Figura 3.2.	Algoritmo genético implementado.....	57
Figura 3.3.	Codificación y puntos de cruce de un individuo.....	59
Figura 3.4.	Ilustración del esquema de representación Doumpos et. Al 2009.....	66
Figura 3.5.	Ilustración del esquema de representación modificado.....	68
Figura 3.6.	Algoritmo evolutivo diferencial.....	70
Figura 3.7.	Diagrama a bloques del algoritmo evolutivo diferencial	71
Figura 3.8.	Codificación del individuo en el algoritmo evolutivo diferencial	72
Figura 3.9.	Generador de instancias para experimentación del método PDA....	74
Figura 4.1.	Implementación del algoritmo multiobjetivo MOEA/D sin el método PDA.....	81
Figura 4.2.	Inserción de la población semilla en la población inicial de MOEA/D.....	84
Figura 4.3.	Generador de instancias para experimentación del algoritmo MOEA/D.....	85

Lista de símbolos y acrónimos

<i>A</i>	Conjunto de objetos a clasificar
AG	Algoritmo Genético
AHP	Proceso Analítico Jerárquico
<i>B</i>	Conjunto de perfiles frontera
CSP	Problema de Clasificación con Restricciones
DIS-CARD	Tamaños Deseados de Clases
DM	Decisor
ED	Evolución Diferencial
ELECTRE	Eliminación y Elección Expresando la Realidad
EP	Población Externa
<i>G</i>	Familia coherente de criterios
<i>h</i>	Número de objetos de referencia, donde $ T = h$
<i>m</i>	Número de criterios, donde $ G = m$
<i>M</i>	Número de categorías
MAUT	Teoría de Utilidad Multi-Atributo
MCDA	Análisis Multicriterio para la Toma de Decisiones
MILP	Programación Lineal Entera Mixta
MOEA	Algoritmo Evolutivo Multiobjetivo
MOEA/D	Algoritmo Evolutivo Multiobjetivo basado en Descomposición
MOGA	Algoritmo Genético Multiobjetivo
MOP	Problema Multiobjetivo
<i>n</i>	Número de objetos a clasificar, donde $ A = n$
NPGA	Algoritmo Genético de Nicho de Pareto
NSGA	Algoritmo Genético de Ordenamiento No Dominado
OC	Objetos Correctos
OD1	Objetos con Diferencia de 1 categoría
<i>p</i>	Número de perfiles frontera, $ B = p$
<i>P</i>	Umbral de Preferencia
PDA	Análisis de Desagregación de Preferencias
PROMETHEE	Método de Organización de Ordenamientos de Preferencias para el Enriquecimiento de las Evaluaciones

Q	Umbral de Indiferencia
SMART	Técnica de Clasificación de Múltiples Atributos Simples
T	Conjunto de objetos de referencia
U	Umbral de Pre-veto
V	Umbral de Veto
V_T	Violaciones al conjunto de referencia
$V_{C_1} \dots V_{C_M}$	Violaciones al tamaño de cada una de las M categorías
VEGA	Algoritmo Genético para el Vector Evaluado
W	Pesos de los criterios
λ	Lambda, umbral de credibilidad

Resumen

Este trabajo aborda un problema de clasificación ordinal en el que los objetos son evaluados por múltiples criterios. La investigación se centra en el problema de clasificación con restricciones al tamaño de las categorías (clases), cuando el encargado de la decisión (decision-maker, DM) siente la necesidad o la conveniencia de establecer restricciones sobre el número de objetos que deben ser asignados a las diferentes categorías. La propuesta utiliza un conjunto de ejemplos de entrenamiento (ejemplos de asignación) que contiene información implícita de la política de asignación del DM. Se aplica un Análisis de Desagregación de Preferencias (PDA) (obtención indirecta de los parámetros del modelo de preferencias) al conjunto de referencia, para inferir un conjunto de parámetros preferenciales completo (pesos de los criterios, umbrales de discriminación y de veto, perfiles frontera y umbral de credibilidad) de un modelo ELECTRE TRI-B. Con esta información preferencial (modelo) y resolviendo un problema de optimización multiobjetivo con el algoritmo MOEA/D, se logra encontrar la mejor forma de asignar las acciones del conjunto A entre las categorías. El modelo preferencial obtenido reproduce la política de asignación del DM implícita en el conjunto de referencia, considerando las restricciones sobre los tamaños de las categorías

Palabras clave: Análisis multicriterio para la toma de decisiones, clasificación ordinal, restricción al tamaño de las categorías, relaciones de sobreclasificación valuada, algoritmos evolutivos multiobjetivos, MOEA/D.

Abstract

This work addresses the ordinal classification (sorting) problem in which the objects (actions) are evaluated with multiple criteria. In this kind of problem the classes (categories) are ordered. This research focuses on problems with category size restrictions, in which the decision maker (DM) wants to fulfill some constraints on the number of objects that should be assigned to certain categories. The proposal uses a set of training examples (assignment examples), that contains implicit information about the DM's assignment policy. A Preference Disaggregation Analysis (PDA) (indirect parameter elicitation) is performed on the set of training examples and plausible ELECTRE TRI-B complete model's parameters (weights of criteria, discrimination and veto thresholds, reference profile and credibility threshold) are inferred. Using this preferential information and solving a multiobjective optimization problem with the MOEA/D algorithm, it is possible to achieve the best way to assign the actions from set A between categories. The model is consistent with the assignment policy and the size restrictions from the DM.

Keywords: Multiple criteria decision analysis, ordinal classification, category size constraints, Valued outranking relations, Multiobjective evolutionary algorithms, MOEA/D.

Capítulo 1.

Introducción

1.1. Generalidades.

La toma de decisiones es un proceso mental, llevado a cabo para la selección de un curso de acción (alternativa, acción potencial), de un conjunto de alternativas de decisión. Este es un proceso mental llevado a cabo por un tomador de decisiones (DM, Decision Maker). Tomamos decisiones en casa, trabajo, deporte, en todas las esferas de la vida. Ante algún cuestionamiento, elegimos: sí, no; opción uno, opción dos; azul, rojo; etc., la mayor parte de las veces tomando en cuenta uno, varios, incluso muchos criterios, basándonos en nuestras preferencias, experiencias o instintos. Sin embargo, en los campos de la ingeniería, negocios, política y ciencias, las decisiones pueden involucrar grandes cantidades de dinero y/o tener impacto directo en el bienestar de la sociedad. Además, la toma de decisiones en estos campos suele ser compleja ya que se cuenta con una buena cantidad de opciones y criterios a considerar que normalmente se encuentran en conflicto entre sí.

Habitualmente se considera un conjunto A de opciones potenciales, que se puede presentar en distintas formas, y que constituye el objeto del problema de decisión en cuestión.

Los desempeños de cada uno de los objetos del conjunto $A = \{a_1, a_1, \dots, a_n\}$, pueden ser caracterizados tomando en cuenta un solo criterio o múltiples criterios. Cuando se utiliza un solo criterio definido por $G = \{g_1\}$, es relativamente fácil para el DM decidir cuál o cuáles alternativas son elegidas al considerar un problema del tipo

$$\text{optimizar}\{g_1(a)|a \in A\}$$

Por otro lado, cuando se consideran simultáneamente un conjunto de criterios de evaluación $G = \{g_1, g_2, \dots, g_m\}$, el DM se enfrenta a una situación de mayor complejidad (Figueira et al., 2005), que se puede representar como

$$\text{optimizar}\{g_1(a), g_2(a), \dots, g_m(a)|a \in A\}$$

expresión que refleja el carácter conflictivo del conjunto de criterios de evaluación, ya que es posible que exista una alternativa cuyo desempeño en ciertos criterios supere a otras alternativas, pero al mismo tiempo esta alternativa sea superada en otros criterios por otras alternativas. La tabla 1.1 muestra cuatro alternativas evaluadas en 4 criterios, en ella se puede observar que la alternativa 1 supera a las alternativas 2, 3 y 4 en el criterio 3 ; pero es superada por las alternativas 2,3 y 4 en el criterio 1. La selección de la mejor alternativa no es un proceso obvio en lo absoluto, ni siquiera está bien definido matemáticamente; depende del sistema de preferencias y prioridades del decisor. Para que una alternativa domine a otra, (solo en este caso la selección es independiente de las preferencias del decisor), se requiere que sea mejor o igual en todos los criterios y mejor en al menos uno de los criterios. En el ejemplo de la tabla 1.1 no hay ninguna alternativa que domine al resto, las 4 alternativas son no dominadas.

Usualmente no existe una alternativa que optimice todos los criterios al mismo tiempo, en esos casos se debe elegir la alternativa con el mejor compromiso de solución de acuerdo a las preferencias del decisor. La mayoría de los problemas de decisión tienen esta naturaleza multicriterio.

Tabla 1.1 Ejemplo de alternativas no dominadas

	Criterio 1	Criterio 2	Criterio 3	Criterio 4
Alternativa 1	0.8	1.5	2.4	1.0
Alternativa 2	1.3	1.2	2.2	2.7
Alternativa 3	1.2	2.1	1.9	2.1
Alternativa 4	1.6	1.9	2.1	1.4

Los problemas de decisión considerados en el Análisis Multicriterio para la Toma de Decisiones (MCDA) tratan un conjunto finito de alternativas evaluadas sobre una familia de criterios coherentes. MCDA da al DM herramientas y métodos para estructurar el problema, comparar y evaluar alternativas, para apoyar en el proceso de tomar una decisión.

Según la escuela Europea de decisión multicriterio, al considerar un problema de toma de decisiones discreto, hay cuatro tipos de análisis que pueden llevarse a cabo para proporcionar apoyo significativo al DM (Roy, 1985). Estos tipos de análisis se muestran en

la figura 1.1 (cf. Doumpos y Zopounidis, 2004) ellos son:

1. La descripción. Este análisis consiste en identificar las características más notables de las alternativas y realizar una descripción de ellas basándose en sus características.
2. Problema de selección ($P.\alpha$). Identificar la mejor alternativa o seleccionar un conjunto limitado de las mejores alternativas.
3. Problema de clasificación ($P.\beta$). Consiste en asignar cada alternativa en categorías predefinidas. Estas categorías corresponden a un conjunto de M clases, donde M es un conjunto de etiquetas.
4. Problema de ordenamiento ($P.\gamma$). Construir un ordenamiento (ranking) de las alternativas de la mejor a la peor. La prescripción resultante puede ser dada en términos de un orden parcial o de un orden completo.

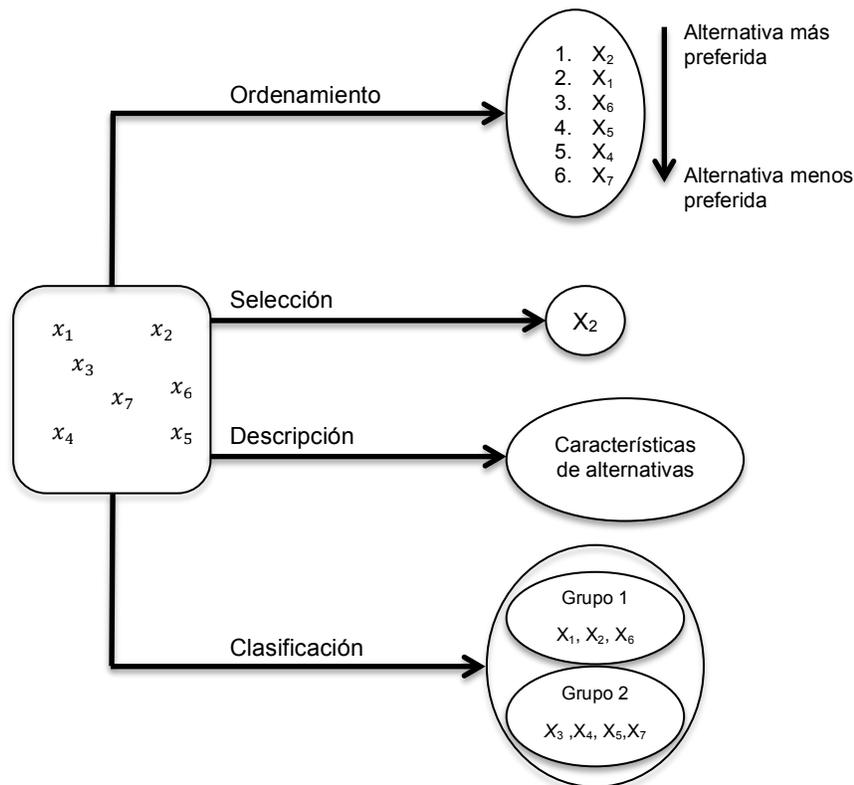


Figura 1.1. Problemas de la toma de decisiones

Un problema de clasificación ($P.\beta$) surge cuando se tienen que asignar acciones de un conjunto A , en categorías que son definidas en términos del destino final o el propósito de las acciones. Se admite que el conjunto A puede evolucionar (agregando o quitando acciones).

Esta problemática lleva a recomendaciones de asignación de acciones en las categorías, por medio de una metodología de asignación automática, que puede ser utilizada repetidamente.

El conjunto de categorías debe ser definido a priori. La definición de cada categoría es basada en el hecho de que ésta debe ser concebida antes de que se asignen las acciones. En los problemas de clasificación, cada acción es considerada de manera independiente con respecto a las demás acciones para determinar a cuál categoría debe ser asignada. La asignación se realiza por medio de comparaciones con perfiles (fronteras, límites), normas o referencias. El resultado de la asignación es expresado en términos de noción absoluta de “asignado” o “no asignado” a una categoría, “similar” o “no similar” a un perfil de referencia, “adecuado” o “no adecuado” a alguna norma.

El problema de clasificación trata de juicios absolutos, consiste en asignar cada acción a una de las categorías predefinidas. La asignación de una acción a resulta de la evaluación intrínseca de a en todos los criterios de evaluación. La asignación de a a una categoría específica, no influencia la categoría a la cual otra acción b debería ser asignada.

Un caso especial de los problemas de clasificación son los problemas de clasificación con restricciones en el tamaño de las categorías. Un Problema de Clasificación con Restricciones surge de situaciones de decisión formuladas en un problema de clasificación en el cual es necesario introducir especificaciones en los tamaños de las categorías. Estas especificaciones son las restricciones sobre el tamaño de las categorías.

1.1.1. Enfoques principales del análisis de la decisión

El análisis matemático en la teoría de la decisión proporciona dos enfoques principales para construir un modelo global de preferencia de un actor envuelto en el proceso de decisión:

1) El enfoque normativo - funcional basado en el axioma normativo de comparabilidad perfectamente transitiva (Fishburn, 1970). En 1944 Von Neumann y Morganstern

axiomatizaron la teoría de utilidad esperada y así asentó las bases de la Teoría de la Utilidad Multi-atributo (MAUT), con gran aplicación en econometría (Von Neumann y Morganstern, 1944). Estos métodos parten del supuesto de que el decisor trata de maximizar una función de utilidad que agrega los distintos criterios que intervienen en el problema. Cuando el problema es discreto y no existe una situación de incertidumbre, esta función se denomina función de valor.

MAUT (Multiple Attribute Utility Theory) asume que un problema de decisión puede modelarse mediante funciones de valor reales que pueden ser maximizadas. Los métodos basados en la función de valor consisten en construir una función (v) que asocia un número real a cada una de las alternativas posibles. Este número refleja el valor o la utilidad que cada alternativa tiene para el decisor. La principal dificultad de estos métodos consiste precisamente en encontrar dicha función de valor, pero una vez obtenida, el problema de decidir la mejor de las alternativas se reduce a obtener el máximo/mínimo de todos los valores calculados (French, 1986).

Basándose en la existencia de la función valor la escuela americana propone varios métodos prácticos, siendo los métodos más importantes: la suma ponderada (French, 1986), el método de las Jerarquías Analíticas o Proceso Analítico Jerárquico (AHP) (Saaty, 1980), y el método SMART (Edwards y Barron, 1994) (The Simple Multi-attribute Rating Technique). El método de clasificación multicriterio UTADIS (UTilités Additives DIScriminantes) propuesto por Devaud (1980) es el enfoque principal de la escuela americana para resolver problemas de clasificación multicriterio. Es un método que asigna alternativas a una de las categorías ordenadas y predefinidas. UTADIS requiere la disponibilidad de algunos ejemplos de asignación de acciones en las categorías, hechas por el decisor. A partir de esa información, se crea un modelo de función lineal por tramos sobre la base del cual se asignan las acciones.

Los métodos de funciones de valor de la escuela americana se destacan por una teoría axiomática sólida, que modela la racionalidad, basada en relaciones de preferencia y un grupo de axiomas en el que se destaca el de “comparabilidad transitiva”. Una desventaja de

estos métodos es que el modelo de agregación es compensatorio, es decir, la pérdida en un criterio puede ser compensada por una ganancia en otro criterio.

2) El enfoque relacional con el trabajo de Bernard Roy en los años 70 (Roy, 1991) y la contribución de varios científicos europeos, fue el fundamento de la metodología de Ayuda a la Decisión Multicriterio por reflejar una actitud dentro de la línea del pensamiento constructivista. Esta familia de métodos persigue ayudar al decisor a resolver el problema teniendo en cuenta las dificultades para la construcción de la función de valor del enfoque de la escuela americana. Se distinguen por su importancia los métodos ELECTRE (Roy, 1985)(The ELimination Et Choix Traduisant la REalité) y los métodos PROMETHEE (Brans y Vincke, 1985)((Preference Ranking Organization Methods for Enrichment Evaluations), métodos también llamados de no-inferioridad. Estos métodos constituyen el núcleo de lo que se conoce en la literatura como el enfoque de “outranking”.

El método PROMETHEE desarrollado por Brans y Vincke en 1985 consiste en la construcción de relaciones de preferencia “borrosas”, incorporando conceptos y parámetros que poseen alguna interpretación física o económica fácilmente comprensibles por el decisor. PROMETHEE hace uso del concepto de pseudocriterio ya que construye el grado de preferencia entre cada par de acciones ordenadas, tomando en cuenta la diferencia de comportamiento de esas acciones respecto a cada atributo. La evaluación de esas diferencias se realiza mediante ciertas funciones de preferencia escogidas por el decisor, quién además debe proporcionar los umbrales de indiferencia y de preferencia asociados a estos pseudocriterios

Otras variantes del método abordan situaciones más sofisticadas de decisión, en particular problemas con un componente estocástico. Así se han desarrollado las versiones PROMETHEE II, PROMETHEE III, PROMETHEE IV y PROMETHEE V. En PROMETHEE V, Brans y Mareschal (1990) incorporan una filosofía de optimización entera a efectos de abordar problemas de selección de inversiones con restricciones presupuestarias.

Los métodos ELECTRE construyen relaciones de “outranking” entre pares de acciones, que, al ser convenientemente explotadas, permiten resolver problemas de clasificación

ordinal, selección y ranking. En la actualidad han sido desarrollados los procedimientos ELECTRE I, II, III, IV, IS, ELECTRE TRI-B, ELECTRE TRI-nC y ELECTRE TRI-nB, los que brindan procedimientos para resolver diferentes tipos de problemas suscitados en el tratamiento de la teoría de la decisión.

La palabra “outranking” es la traducción en inglés de la palabra en francés “surclassement”, utilizada inicialmente por Bernard Roy en sus trabajos de investigación en la década de los 60. Una relación del tipo “outranking” es una relación binaria que establece la preferencia entre dos objetos (Roy, 1996). Se lee “es al menos tan bueno como”. En algunos trabajos de investigación, en la traducción al español de la palabra “outranking” se utiliza la palabra “sobre-clasificación”. También se utiliza la palabra “superación”, sin embargo, el utilizar esta palabra infiere que la relación de “outranking” es antisimétrica, cuando no lo es. En este trabajo utilizaremos el término “no-inferioridad”.

Las preferencias en los métodos ELECTRE son modeladas usando relaciones binarias de no-inferioridad (outranking relations), S , cuyo significado es “al menos tan bueno como”. Considerando dos acciones a y b , cuatro situaciones pueden ocurrir:

- aSb y no $bSa \Rightarrow aPb$, a es preferido estrictamente a b .
- bSa y no $aSb \Rightarrow bPa$, b es preferido estrictamente a a .
- aSb y $bSa \Rightarrow aIb$, a es indiferente a b .
- No aSb y no $bSa \Rightarrow aRb$, a es incomparable a b .

Cuando se utilizan relaciones de no-inferioridad el modelo de preferencias introduce la relación de preferencia de incomparabilidad R . Esta relación es muy útil cuando el DM y/o el analista no son capaces de comparar dos acciones.

La construcción de una relación de no-inferioridad es basada en dos conceptos principales:

1. Concordancia. Para que la aserción aSb sea validada, una mayoría suficiente de criterios debe estar a favor de esta aserción.
2. Discordancia. Cuando la condición de concordancia se mantiene, ninguno de los criterios de la minoría debe oponerse muy fuerte a la aserción aSb .

Para que la aserción aSb sea validada, se deben cumplir estas dos condiciones.

Los métodos ELECTRE constan de dos procedimientos principales: construcción de una o varias relaciones de no-inferioridad, seguidas por un procedimiento de explotación.

El procedimiento de construcción ayuda a comparar cada par de acciones en una manera comprensiva. El procedimiento de explotación es utilizado para elaborar recomendaciones dependiendo de la problemática a resolver (selección, ordenamiento o clasificación). Cada método es caracterizado por sus procedimientos de construcción y explotación.

El método ELECTRE TRI-B es el método más importante para resolver problemas de clasificación ordinal en la familia de métodos ELECTRE. ELECTRE TRI-B es un método de clasificación multicriterio que asigna acciones a categorías predefinidas, donde las categorías son separadas por perfiles de frontera. La asignación de una acción resulta de la comparación de la acción con los perfiles que definen los límites de las categorías.

El método ELECTRE TRI-B es el método de clasificación que más aplicaciones tiene documentadas en la literatura (eg. Mousseau et al., 1999; Mousseau et al., 1999; Gomes et al., 2004; Gomes et al., 2007; Doumpos et al., 2009; Lu et al., 2010; Fontana et al., 2013; Norese et al., 2014) y recientemente han surgido otros métodos que utilizan mas información de referencia como el método ELECTRE TRI-C (Almeida-Dias et al., 2010), ELECTRE TRI-nC (Almeida-Dias et al., 2012) y ELECTRE TRI-nB (Fernández et al. 2017).

El método ELECTRE TRI-B, al igual que todos los métodos basados en ELECTRE, tienen la capacidad de manejar información ordinal, cualitativa y con cierto grado de imprecisión, además de preferencias intransitivas y situaciones de veto e incomparabilidad. Su capacidad para manejar información no-cardinal y efectos de veto es una característica muy importante de ELECTRE que no poseen otros enfoques de ayuda a la decisión.

Este trabajo estará basado en el enfoque de la escuela Europea para la propuesta del modelo que dará solución al problema de clasificación con restricción en el tamaño de las categorías. Como parte de la propuesta se utilizará el método ELECTRE TRI-B como herramienta para sugerir asignaciones de objetos a categorías.

1.2. Antecedentes

Este trabajo de investigación aborda un problema de clasificación multicriterio donde las categorías y los criterios tienen una escala al menos ordinal, es decir, donde se distinguen las preferencias del decisor para valores distintos en sus dominios. Ejemplos clásicos de este tipo de problemas son los siguientes: clasificación de países con diferentes niveles de riesgo basados en sus criterios sociopolíticos y económicos, clasificación de aplicaciones de crédito bancarios, clasificación de solicitudes de ingreso de estudiantes y la clasificación de diferentes tipos de fondos de financiamiento. Algunos problemas de clasificación son mostrados en Doumpos y Zopounidis (2004).

Se presentan situaciones donde se requiere imponer restricciones acerca de cómo deben ser distribuidos los objetos entre las categorías. Mousseau (2003) define este tipo de problemas como Problemas de Clasificación con Restricciones (CSP, por sus siglas en inglés Constrained Sorting Problem) con respecto al tamaño de la categoría. Los problemas CSP pueden surgir en una gran variedad de situaciones reales, por ejemplo:

1. Una agencia de financiamiento puede tener restricciones acerca del número de proyectos que puede financiar en un alto nivel.
2. En el proceso de admisión de estudiantes se presentan restricciones similares cuando el número de estudiantes que pueden ser aceptados es menor al número de solicitudes de admisión que son recibidas por la universidad.
3. En financiamiento públicos, por ejemplo de proyectos de investigación, donde el presupuesto solicitado es mayor al presupuesto disponible.

En la tabla 1.2 se listan los escasos estudios en la literatura que consideran la clasificación con restricciones en los tamaños de las categorías. Mousseau (2003) propone un método basado en UTADIS para inferir los parámetros preferenciales del modelo de ejemplos de asignación propuestos por un DM y las especificaciones acerca de los tamaños de las categorías. Resuelve un problema con un conjunto 100 objetos concernientes a aplicaciones de crédito. El decisor asigna 8 de los 100 objetos en las 3 categorías, que son utilizados como ejemplos de asignación. UTADIS ayuda a inferir los parámetros preferenciales que

ubiquen mejor a los ejemplos de asignación y respeten los tamaños de las categorías solicitados. Modelan la información preferencial de los ejemplos de asignación y las restricciones a los tamaños de las categorías como restricciones mediante programación lineal entera mixta (MILP).

Cardinal y otros (2011) resuelven un problema de solicitudes de admisión a la especialidad de Ingeniería Industrial de la Escuela de Ingeniería Ecole Centrale Paris, donde el decano establece cuantos estudiantes pueden ser admitidos por especialidad en la escuela. Cada año el número de aplicantes es mayor a esta cantidad. Abordan el problema como un problema de cartera. Utilizan una versión simplificada del método de no-inferioridad ELECTRE TRI-B, evitan los umbrales de preferencia e indiferencia. La información preferencial de los 5 ejemplos de asignación, las 3 declaraciones del decisor y las restricciones a los tamaños de las categorías, es modelada como restricciones mediante programación lineal entera mixta MILP.

Un problema de selección de cartera con restricción en el tamaño de las categorías fue resuelto por Zheng y otros (2011). Utilizan el método ELECTRE TRI-B para clasificar las acciones en las categorías. Consideran una versión simplificada de ELECTRE TRI-B que ignora los umbrales de preferencia, indiferencia y veto, versión que corresponde al estudio axiomático de Bouyssou y Marchant (2007a y 2007b). Su metodología se basa en dos etapas: en una primera etapa realizan un proceso de inferencia de parámetros a 30 ejemplos de referencia para obtener un modelo de preferencias apropiado. Los parámetros inferidos son: pesos de los criterios, perfiles frontera y umbral de credibilidad. En la segunda etapa se ajustan los parámetros inferidos para clasificar a las 100 acciones en las categorías, respetando los tamaños de las categorías requeridos.

Kadzinski y Słowiński (2013) muestran una propuesta de desagregación de preferencias para un método llamado DIS-CARD, que aborda el problema de clasificación ordinal con restricciones a los tamaños de las categorías. El método DIS-CARD utiliza ejemplos de ordenamiento proporcionados como ejemplos de asignación, y modela esta información preferencial y las restricciones a los tamaños de las categorías como restricciones mediante programación MILP. Finalmente adaptan el modelo MILP a dos tipos de modelos preferenciales: a una función de valor (UTADIS) y a una relación de no-inferioridad

(ELECTRE TRI-B). Modelan la información preferencial de los ejemplos de referencia y los tamaños de las categorías como restricciones y resuelven el modelo mediante programación entera mixta.

Tabla 1.2 Publicaciones asociadas a los problemas CSP

Publicación	Año	No. de categorías	No. de Criterios	No. de objetos por clasificar (A)	No. de Objetos de entrenamiento (T)	Método de clasificación	Técnica de solución
On the notion of category size in multiple criteria sorting models, Mousseau et al., 2003.	2003	3	7	100	8 ejemplos de asignación	UTADIS	MILP
Constrained multicriteria sorting method applied to portfolio selection, Zheng et al., 2011.	2011	3	6	100	30	ELECTRE TRI-B	MILP
An application of constrained multicriteria sorting to student selection, Cardinal et al., 2011.	2011	4	6	76	5 y 3 declaraciones del DM	ELECTRE TRI-B	MILP
DIS-CARD: a new method of multiple criteria sorting to classes with desired cardinality, Kadsinski y Slowinski, 2013.	2013	4	3	15	4	DIS-CARD	MILP
Parametric evaluation of research units with respect to reference profile, Kadsinski y Slowinski, 2015.	2015	3	4	20	--	PROMETHEE	MILP
Multicriteria sorting with category size restrictions, Köksalan et al., 2017.	2017	5	3	81	2,3,5,7 objetos por categoría	UTADIS	MILP
		5	7	100	2,3 objetos por categoría		
An interactive algorithm for multiple criteria constrained sorting problema, Özpeynirci et al., 2017.	2017	4	8	76	6	UTADIS MR-Sort	MILP
		2	6	76	3		
Metodología propuesta							
Un nuevo enfoque para resolver el problema de clasificación ordinal con restricciones al número de objetos en las categorías. Fernández et al, 2018.	2018	3	3	50	3,5,7,10 objetos por categoría	ELECTRE TRI-B	MOEA
		4	5	100			
		5	7	200 1000			

Kadzinski y Słowiński (2015) resuelven un problema de asignación de presupuesto a unidades de investigación para el Ministerio de Ciencia y Educación Superior de Polonia. Desarrollan un método que considera clasificación y ordenamiento multicriterio para generar su modelo de solución y resolverlo mediante un algoritmo MILP. El modelo es generado mediante un método de no-inferioridad (outranking) inspirado en PROMETHEE.

Özpeynirci y otros (2017) resuelven dos problemas CSP incorporando interactividad con el DM para ayudar al algoritmo en la búsqueda de la solución. Para resolver los problemas utilizan los métodos UTADIS y MR-Sort, y resuelven el modelo propuesto con programación lineal entera mixta (MILP).

Como puede verse en la tabla 1.2, los trabajos antecedentes se dividen en dos grupos: i) los que emplean UTADIS, basado en información cardinal y en un modelo totalmente compensatorio, con preferencias transitivas y sin posibilidades de veto; e ii) los que se apoyan en métodos de no-inferioridad como ELECTRE TRI-B, PROMETHEE y MR-SORT (una simplificación de ELECTRE TRI-B). Los del segundo tipo prescinden del poder de veto para evitar la complejidad matemática del problema de optimización que resuelven, y permitir el empleo de MILP.

Todos los trabajos realizados con anterioridad que infieren los parámetros preferenciales utilizan la información preferencial de referencia (ejemplos de asignación) como una verdad absoluta que debe ser respetada. Las modelaciones hechas sólo permiten flexibilidad sobre las preferencias expresadas sobre las restricciones a los tamaños de las categorías. De tal manera que las soluciones propuestas tratan de ajustar lo mejor posible los parámetros preferenciales de los modelos generados, respetando la información preferencial de entrenamiento y buscando cumplir con las restricciones en los tamaños de las categorías impuestas.

Esta investigación se centra en resolver el problema de clasificación con restricción en el tamaño de las categorías como un problema multiobjetivo empleando el método de clasificación ELECTRE TRI-B y la inferencia de parámetros (Preference Disaggregation Analysis, PDA) para extraer la política de asignación del DM, implícita en un conjunto de ejemplos de referencia. Se admite la capacidad de veto y las incomparabilidades que de ella

resultan, y la no-linealidad se maneja empleando algoritmos evolutivos. Con el enfoque multiobjetivo se busca eliminar la consideración de restricción absoluta de las preferencias implícitas en el conjunto de entrenamiento; es decir, mediante su enfrentamiento a un problema de optimización multiobjetivo se pretende que el decisor aprenda sobre el problema, y tenga la posibilidad de reconocer posibles inconsistencias en el conjunto de entrenamiento. De tal manera que dentro de las soluciones obtenidas puede haber opciones que asignen algunos objetos de referencia de manera distinta al conjunto de entrenamiento, permitiendo con esto, cumplir mejor con las restricciones a los tamaños de las categorías.

Por lo anterior, pretendemos que el nuevo método posea ventajas competitivas respecto a sus antecedentes, derivadas de a) el manejo de situaciones de incomparabilidad y veto, así como de información ordinal y cualitativa; b) mayor flexibilidad en el tratamiento de las restricciones al tamaño de las categorías, y al valor no absoluto de la información presente en el conjunto de objetos de entrenamiento. Al mismo tiempo, y es otro aspecto diferenciante de los trabajos antecedentes, pretendemos realizar un estudio computacional exhaustivo, en un rango amplio de las variables que determinan la dimensión del problema de decisión, de manera que nuestras conclusiones puedan aspirar a ser consideradas generales.

1.3. Descripción del problema

Considere que se tienen dos conjuntos de objetos: un conjunto T de referencia que contiene h objetos de los cuales es conocida su clasificación; y un conjunto A de n objetos cuya clasificación es desconocida. Los objetos de los dos conjuntos están evaluados por un conjunto G de criterios coherentes. Además el tomador de decisiones impone restricciones sobre el número de objetos que debe contener cada categoría cuando los objetos del conjunto A sean clasificados.

Considerando esta información de entrada, el problema consiste en clasificar los n objetos del conjunto A en M categorías, respetando las restricciones impuestas al número de objetos que debe tener cada una de ellas y la información preferencial implícita en los objetos del conjunto T . La figura 1.2 muestra la problemática planteada.

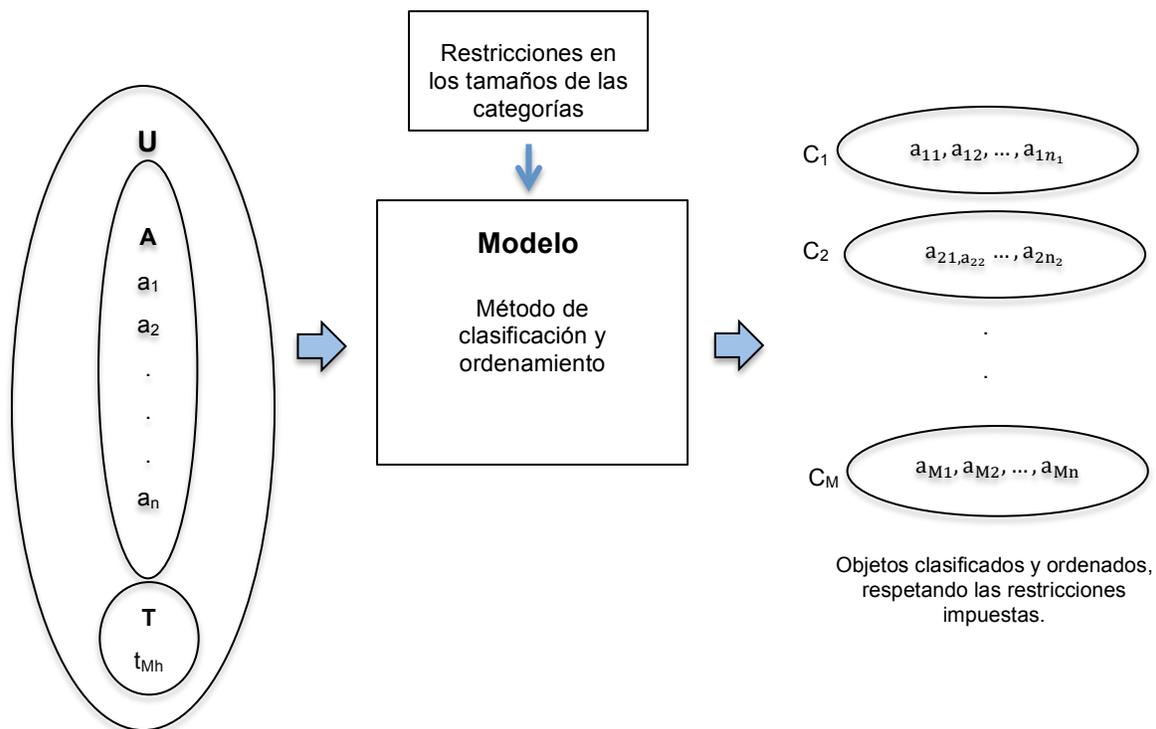


Figura 1.2. Problemática planteada

El problema planteado implica la necesidad de satisfacer las preferencias del decisor, las cuáles están implícitas en el conjunto de referencia y también sus preferencias referentes a los tamaños de las categorías.

Las preferencias impuestas a los tamaños de las categorías brindan mas información al tomador de decisiones si son manejadas de manera individual, es decir, saber con certeza el número de violaciones al tamaño de cada categoría que se tienen con un modelo (conjunto preferencial). Sin embargo, esto trae la necesidad de manejar como un objetivo el número de violaciones a cada categoría, y por tanto nos adentra en un escenario de un problema de optimización multiobjetivo, donde el número de objetivos es $M + 1$, uno para las violaciones de cada categoría y otro para manejar las violaciones a la clasificación de los objetos de referencia.

La propuesta para resolver el problema de clasificación ordinal multicriterio con restricciones en los tamaños de las categorías formulado como un problema de optimización multiobjetivo se muestra en la figura 1.3. La propuesta emplea una técnica de desagregación de preferencias para extraer la información preferencial del DM implícita en un conjunto de referencia (conjunto T). Para resolver el modelo multiobjetivo se utilizará el algoritmo multiobjetivo MOEA/D, tomando como entradas la información preferencial obtenida en el proceso de inferencia de parámetros, las restricciones en los tamaños de las categorías impuestas por el decisor y el conjunto de objetos a clasificar (conjunto A). El método de clasificación ordinal a utilizar es ELECTRE TRI-B.

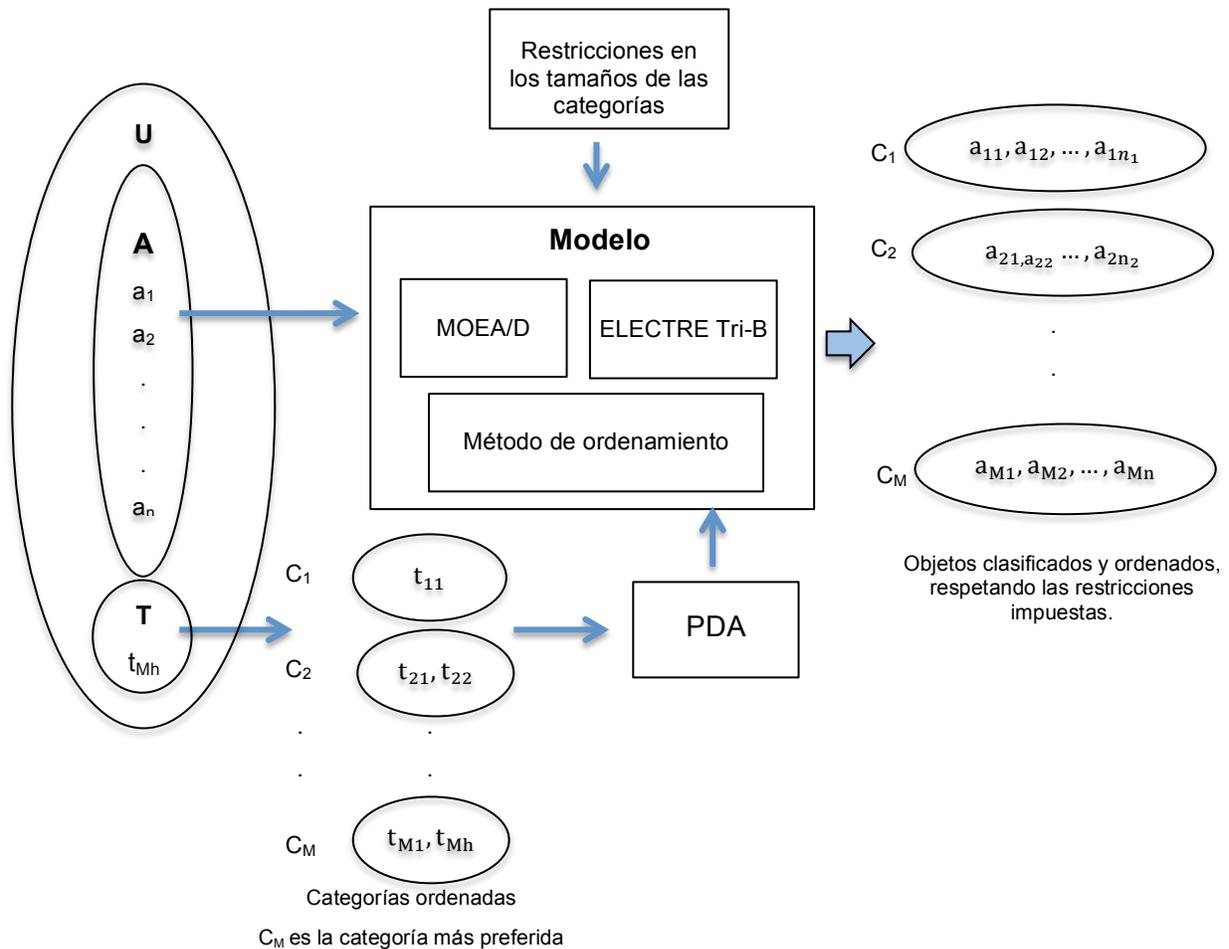


Figura 1.3. Propuesta de solución al problema

Una vez clasificados las n acciones, se aplicará un método de ranking para ordenar las alternativas de cada categoría, de tal manera que sea posible diferenciar entre alternativas

de una misma categoría. La propuesta de solución considera un modelo preferencial completo (Q, P, U, V, W, perfiles frontera y λ) para una mejor caracterización del DM.

1.4. Preguntas de investigación

- ¿Contribuye la optimización multiobjetivo a lograr mejores soluciones desde el punto de vista de las preferencias integrales del DM?
- ¿Ayudan las heurísticas multiobjetivo a resolver el problema de clasificación ordinal multicriterio con restricciones en el tamaño de las categorías con la metodología ELECTRE, utilizando un modelo preferencial completo (Q, P, U, V, W, perfiles y λ) para caracterizar al DM?
- ¿Cuál es la mejor forma de resolver el problema de optimización multiobjetivo?. Resolver primero el problema de extraer las preferencias del DM implícitas en el núcleo de referencia (análisis PDA) y posteriormente incorporar las restricciones a los tamaños de las categorías; o considerar el análisis PDA y las restricciones a los tamaños de las categorías al mismo tiempo.

1.5. Alcance y limitaciones del trabajo de investigación

El alcance de este trabajo de investigación está delimitado al problema de clasificación ordinal multicriterio con restricciones al tamaño de las categorías y a las consideraciones siguientes:

1. El problema debe ser adecuado a la utilización de métodos de clasificación basados en relaciones de no-inferioridad (outranking) con ciertas características como: existencia de condiciones de veto, incomparabilidad, falta de transitividad en las preferencias del decisor e información ordinal en algunos criterios.
2. La experimentación en este trabajo esta limitada a problemas que consideran 3, 5, 7 y 10 objetos por categorías en el conjunto de referencia; desempeño de los objetos evaluados en 3, 5 y 7 criterios; 3, 4 y 5 categorías. Otros valores de estos parámetros

deberán ser abordados por otros trabajos de investigación.

3. La metodología propuesta no ha sido aplicada a casos reales, solo a datos generados artificialmente en donde se sabe que existe al menos una solución (modelo de preferencias).

1.6. Objetivos

En este apartado se presentan los objetivos de este trabajo de investigación, los cuáles tienen como base a las preguntas de investigación planteadas con anterioridad.

1.6.1. Objetivo General

Hacer un aporte original en la modelación y solución de los problemas de clasificación ordinal multicriterio con restricciones sobre el tamaño de las categorías, que mejore las propuestas de modelación y solución actuales.

1.6.2. Objetivos específicos

- Diseñar un modelo original de solución no lineal basado en una propuesta de clasificación-ordenamiento.
- Obtener el modelo de preferencias del conjunto de entrenamiento utilizando una técnica de desagregación de preferencias.
- Resolver el problema de clasificación ordinal multicriterio con restricciones en el tamaño de las categorías, como un problema de optimización multiobjetivo.
- Desarrollar, implementar y aplicar un algoritmo que implemente el modelo propuesto.

1.7. Estructura de la tesis

En el capítulo 2 se presentan los fundamentos teóricos necesarios para el desarrollo de la

tesis. En el capítulo 3 se presenta el Análisis de Desagregación de Preferencias (PDA) implementado con una metaheurística y los resultados experimentales obtenidos. El modelo de clasificación ordinal multicriterio desarrollado se muestra en el capítulo 4. Por último en el capítulo 5, las conclusiones y trabajo futuro.

Capítulo 2.

Fundamentos teóricos

Esta sección contiene la fundamentación teórica necesaria para la realización de este trabajo de investigación. Para abordar el problema de clasificación ordinal con restricción en el número de objetos en las categorías, se requiere la comprensión del método de clasificación ELECTRE TRI-B y se muestra la definición formal de la clasificación con restricción en el tamaño de las categorías. Se estudia el Análisis de Desagregación de Preferencias (PDA) y dos metaheurísticas, algoritmos genéticos y evolución diferencial. Adicionalmente, se muestra el algoritmo de optimización multiobjetivo MOEA/D.

2.1. Definiciones y conceptos básicos

Sea $C = \{C_1, C_2, \dots, C_M\}$ el conjunto finito de categorías ordenadas; ($M \geq 2$); C_M es la categoría más preferida. El término “preferida” es relativo a cada problema de clasificación particular (por ejemplo, más calidad, menos riesgo, más aceptable).

Sea $T = \{t_1, t_2, \dots, t_h\}$ un conjunto finito de ejemplos de asignación (objetos o acciones de referencia), donde $|T| = h$.

Sea $A = \{a_1, a_2, \dots, a_n\}$ un conjunto finito de decisión con objetos o acciones a clasificar, donde $|A| = n$.

Sea $G = \{g_1, g_2, \dots, g_m\}$ una familia coherente de criterios (Bouyssou et al., 2002), donde $g_j(a)$ es la evaluación en el j -ésimo criterio de la acción a , donde $a \in A$, $g_j(a) \in \mathfrak{R}^+$ y $|G| = m$.

Sea $W = \{w_1, w_2, \dots, w_m\}$ el conjunto de pesos, donde $w_j \in (0,1)$ denota la importancia del j -ésimo criterio g_j , donde $g_j \in G$ y $\sum_{j=1}^m w_j = 1$.

Sea $B = \{b_1, b_2, \dots, b_p\}$ el conjunto de perfiles frontera, donde $|B| = p$, $p = M - 1$ y b_k separa las categorías C_k y C_{k+1} .

Los criterios de evaluación deben estar diseñados para capturar la naturaleza multidimensional del desempeño de las alternativas. Al elegir una familia de criterios para evaluar un conjunto de alternativas, está debe ser de un número suficientemente pequeño (legible), debe ser operativa; además, debe contener todos los puntos de vista (exhaustivo), ser monótona y que cada criterio sea contado sólo una vez (no redundancia). Al cumplirse estas reglas, se está definiendo una familia coherente de criterios (Bouyssou, 2002).

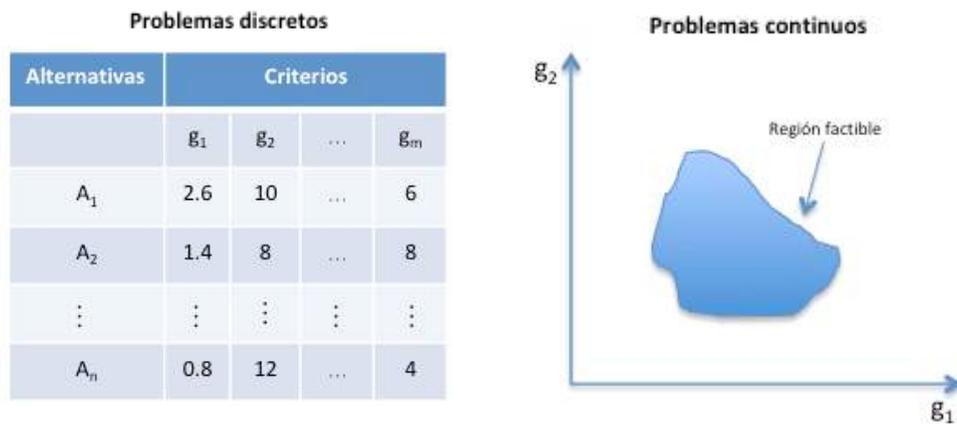


Figura 2.1. Ejemplo de problemas discretos y continuos
Fuente: Elaboración propia con base en (Doumpous y Zopounidis, 2004. p.2).

2.2. Tipo de problema a tratar

Los problemas de toma de decisiones pueden ser clasificados en dos categorías: problemas de decisión discretos y problemas de decisión continuos. En el primer tipo, los objetos (acciones) se describen explícitamente, generalmente en forma enumerativa. En la segunda clase el conjunto de objetos se describe por un conjunto de restricciones; la región factible está formada por un conjunto de gran cardinal, incluso no numerable. Por ello, sólo se puede identificar la región donde se encuentran las alternativas de decisión; conocida como región factible o región solución, en la cual, cada uno de los puntos de dentro de dicha región, concierne a una acción específica (Doumpous, y Zopounidis, 2004). La Figura 2.1

muestra las categorías de los problemas discretos y continuos.

2.3. Problemas continuos

El proceso de optimización de problemas que involucran funciones con restricciones es un tópico fundamental de estudio, ya que la mayoría de problemas del mundo real presentan restricciones de algún tipo. Aunque existen muchos métodos para resolver este tipo de problemas de optimización, es muy difícil encontrar uno que sea siempre eficiente para cualquier tipo de restricciones.

Muchos problemas del mundo real deben ser resueltos considerando un conjunto de restricciones. Esas restricciones limitan el espacio de búsqueda de soluciones a una zona especial que se denomina factible. Dependiendo de las restricciones, la zona factible puede ser única o estar compuesta de varias zonas dispersas en el espacio de búsqueda, lo cual dificulta aún más la solución del problema. Contrario a lo que pudiera pensarse, el uso de restricciones, si bien reduce la región del espacio de búsqueda donde residen las soluciones de interés, no necesariamente vuelve al problema más fácil de resolver. De hecho, lo suele hacer más difícil debido a que se complica el proceso de alcanzar la solución de interés (o sea, el óptimo global). Esto sucede como consecuencia de las irregularidades que las restricciones introducen en el espacio de búsqueda.

2.3.1. Optimización de un objetivo

Optimizar significa encontrar los valores máximos (maximización) o mínimos (minimización) de una función entre todas las soluciones factibles del problema. En problemas con un solo objetivo, existe una única función a optimizar. Aquel individuo (solución en el espacio del problema) que se corresponde con el valor máximo / mínimo del problema, es la solución al problema de optimización. Formalmente,

$$\begin{aligned} \textit{minimizar } F(x) &= f(x) \\ \textit{sujeto a } x &\in \Omega \end{aligned} \tag{2.1}$$

donde Ω es el espacio de decisión, $F: \Omega \rightarrow \mathbf{R}$ consiste de una función objetivo real y \mathbf{R} es llamado el espacio objetivo. El espacio objetivo es definido como el conjunto $\{F(x) | x \in \Omega\}$.

El algoritmo de búsqueda y optimización que se utilice para abordar el problema deberá tener la capacidad de encontrar estos puntos extremos de la función, o al menos soluciones que se correspondan con valores cercanos al óptimo.

2.3.2. Optimización multiobjetivo

La mayor parte de los problemas de optimización del mundo real son naturalmente multiobjetivo. Esto es, suelen tener dos o más funciones objetivo que deben satisfacerse simultáneamente y que posiblemente están en conflicto entre sí. Sin embargo, a fin de simplificar su solución, muchos de estos problemas tienden a modelarse como mono-objetivo usando sólo una de las funciones originales y manejando las adicionales como restricciones. Por ejemplo, con el presupuesto acotado de gastos de supermercado, se intenta comprar los productos necesarios de mayor calidad al menor precio posible. Cuando se decide el lugar para vacacionar, seguramente se consideran cuestiones de distancia al lugar, costos de las diferentes alternativas, opciones de entretenimiento para los diferentes integrantes del grupo, extensión del período de vacaciones, etc. Estos dos ejemplos del mundo real permiten entender la naturaleza básica de la optimización multiobjetivo. Se trata de la búsqueda de opciones considerando la optimización de varios objetivos en simultáneo y normalmente contrapuestos.

Un problema multiobjetivo difiere de un problema mono-objetivo, dado que los primeros requieren la optimización simultánea de más de una función objetivo en paralelo.

La noción de óptimo tiene que ser redefinida en este contexto, donde en lugar de buscar una única solución mejor, se intenta producir un conjunto de buenas soluciones de compromiso. El desafío principal de los algoritmos de optimización multiobjetivo es encontrar este conjunto de soluciones para ofrecer al tomador de decisiones las mejores alternativas entre las disponibles, para que seleccione una de ellas.

Un problema de optimización multiobjetivo (MOP), puede ser formulado como sigue (Zhang et al., 2007):

$$\begin{aligned} \text{minimizar } \mathbf{F}(\mathbf{x}) &= \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{\text{num_obj}}(\mathbf{x}) \right) \\ \text{sujeto a } \mathbf{x} &\in \Omega \end{aligned} \quad (2.2)$$

donde Ω es el espacio de decisión, $\mathbf{F}: \Omega \rightarrow \mathbf{R}^{\text{num_obj}}$ consiste de num_obj funciones objetivo reales y $\mathbf{R}^{\text{num_obj}}$ es llamado el espacio objetivo. El espacio objetivo es definido como el conjunto $\{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \Omega\}$.

En la mayoría de los casos los objetivos en (2.2) se contradicen unos a otros, no hay un punto en Ω que minimiza todos los objetivos simultáneamente, se tiene que encontrar un balance entre ellos. El balance adecuado depende de las preferencias del tomador de decisiones, y por tanto no puede definirse rigurosamente en términos de las características matemáticas del problema.. Sin embargo, las buenas soluciones deben pertenecer a un conjunto que sí tiene una definición matemática precisa: el conjunto de óptimos de Pareto.

Sea $\mathbf{u}, \mathbf{v} \in \mathbf{R}^{\text{num_obj}}$, se dice que \mathbf{u} domina a \mathbf{v} si y solamente si $u_i \leq v_i$ para cada $i \in \{1, \dots, \text{num_obj}\}$ y $u_j < v_j$ para al menos un índice $j \in \{1, \dots, \text{num_obj}\}$. Un punto $\mathbf{x}^* \in \Omega$ es óptimo de Pareto de (2.2) si no hay un punto $\mathbf{x} \in \Omega$ tal que $\mathbf{F}(\mathbf{x})$ domine a $\mathbf{F}(\mathbf{x}^*)$. $\mathbf{F}(\mathbf{x}^*)$ es llamado vector óptimo de Pareto. La mejora de cualquier objetivo en un punto óptimo de Pareto lleva al empeoramiento en al menos uno de los otros objetivos. El conjunto de todos los vectores óptimos de Pareto es llamado el frente de Pareto (PF).

Se dice que una solución a domina a una solución b si, y sólo si (suponiendo maximización):

$$\forall i \in \{1, 2, \dots, k\} | f_i(\mathbf{a}) \geq f_i(\mathbf{b}) \wedge \exists j \in \{1, 2, \dots, k\} | f_j(\mathbf{a}) > f_j(\mathbf{b})$$

Es decir, una solución domina a otra si es mejor o igual en todos los objetivos y mejor en al menos uno de ellos.

Una solución es óptimo de Pareto (solución no dominada) si no es dominada por ninguna otra solución del espacio de soluciones. El conjunto de soluciones no dominadas $\mathbf{X}^* \subset \mathbf{X}$ es

el conjunto de soluciones óptimas de Pareto y compone la solución óptima del problema multiobjetivo.

Normalmente no existe una única solución óptima de Pareto a un problema multiobjetivo, existe un conjunto (en ocasiones infinito) de soluciones no dominadas que forman la frontera de Pareto, ver figura 2.2.

Se puede demostrar que el mejor compromiso (o sea, la solución más apropiada del problema de acuerdo con las preferencias del tomador de decisiones) es necesariamente una solución óptima de Pareto.

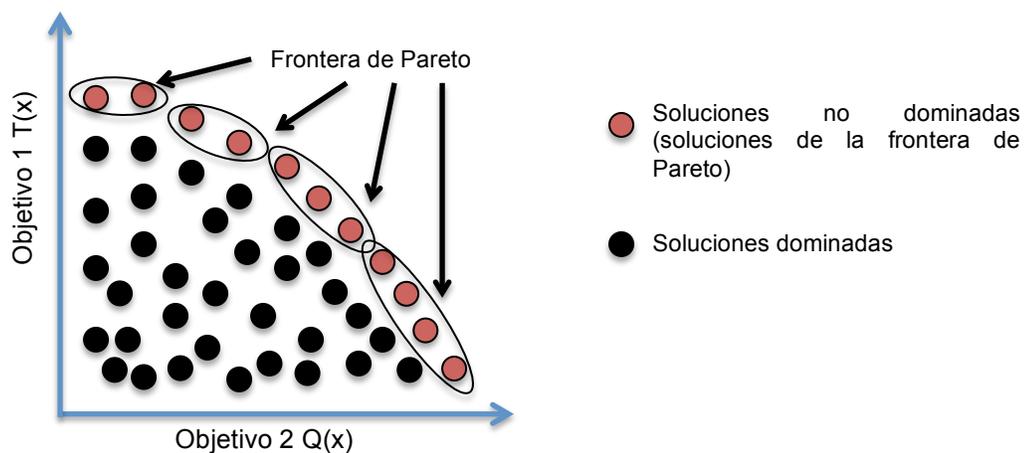


Figura 2.2. Frontera de Pareto para Max $T(x)$ y Max $Q(x)$

Un problema de optimización multiobjetivo puede ser resuelto con métodos convencionales de programación matemática lineal o no lineal, de acuerdo a las características de las funciones objetivo y restricciones del problema. Pero para el tratamiento de la fuerte no-linealidad que conduce a formas inusuales de la frontera de Pareto, los algoritmos evolutivos se han convertido en una herramienta moderna fundamental, con ventajas claras sobre la programación matemática convencional (ver por ejemplo, Coello, 1999).

El potencial de los algoritmos evolutivos para resolver problemas de optimización multiobjetivo se remonta a finales de los 1960s en la tesis doctoral de Rosenberg (1967) donde se indicó la posibilidad de usar algoritmos genéticos en este dominio. Sin embargo, el primer intento real por extender un algoritmo evolutivo a problemas multiobjetivo es el

Vector Evaluated Genetic Algorithm (VEGA) desarrollado por Schaffer en su tesis doctoral de 1984 y presentado en la Primera Conferencia Internacional de Algoritmos Genéticos (en 1985).

Históricamente se cuentan tres generaciones de algoritmos evolutivos multiobjetivo:

1. Primera Generación: Caracterizada por el uso de jerarquización de Pareto y nichos. Algoritmos relativamente simples. También se produjeron enfoques más rudimentarios (por ejemplo, funciones agregativas lineales). Los MOEAs más representativos de esta generación son: NSGA, NPGA y MOGA.
2. Segunda Generación: Comenzó cuando el elitismo llegó a ser un mecanismo estándar. Muchos MOEAs han sido propuestos en la segunda generación, pero muy pocos han sido considerados como un referente. SPEA2 (Zitzler et al., 2001) y NSGA-II (Deb et al., 2002) pueden ser los algoritmos más representativos de esta generación.
3. Tercera Generación: Los algoritmos de primera y segunda generación generan presión selectiva hacia la frontera de Pareto a partir de explotar el concepto de dominancia, y pierden efectividad cuando se incrementa la dimensión del espacio objetivo. En los últimos años con el aumento en la complejidad de los problemas que requieren una mayor cantidad de funciones objetivo a resolver, resultan de interés los MOEAs basados en otros paradigmas. Los MOEAs más representativos de esta generación son los basados en descomposición (MOEA/D y MOEA/D-DE).

Para resolver el problema de optimización multiobjetivo en la sección 4, se utilizará el algoritmo multiobjetivo basado en descomposición MOEA/D.

2.4. Problemas Discretos

Doumpos, y Zopounidis (2004) y Roy (1985), sustentan que dentro de los problemas discretos, existen cuatro tipos de análisis o problemas de decisión, que pueden apoyar al DM en su tarea de realizar su decisión. De esos problemas, Figueira et al. (2010) consideran como principales, a los tres últimos de la lista siguiente:

- Identificarlas y describirlas por sus características.

- Ordenarlas en orden de la mejor a la peor (ranking).
- Seleccionar las mejores (choice).
- Clasificarlas en grupos homogéneos (sorting).

Este trabajo aborda un problema de clasificación ordinal multicriterio donde las categorías y los criterios tienen una escala al menos ordinal, es decir, donde se distinguen las preferencias del decisor para valores distintos en sus dominios. Ejemplos típicos de este tipo de problemas incluyen clasificación de países con diferentes niveles de riesgo basados en sus criterios sociopolíticos y económicos, clasificación de aplicaciones de crédito bancarios, clasificación de solicitudes de ingreso de estudiantes y la clasificación de diferentes tipos de fondos de financiamiento. Algunos problemas de clasificación son mostrados en Doumpos y Zopounidis (2004).

Se presentan situaciones donde se requiere imponer restricciones al problema de clasificación acerca de cómo deben ser distribuidos los objetos entre las categorías. Mousseau (2003) define este tipo de problemas como Problemas de Clasificación con Restricciones (CSP, por sus siglas en inglés Constrained Sorting Problem) con respecto al tamaño de la categoría. Los problemas CSP pueden surgir en una gran variedad de situaciones reales, por ejemplo:

1. Una agencia de financiamiento puede tener restricciones acerca del número de proyectos que puede financiar en un alto nivel.
2. En el proceso de admisión de estudiantes se presentan restricciones similares cuando el número de estudiantes que pueden ser aceptados es menor al número de solicitudes de admisión que son recibidas por la universidad.
3. En financiamiento público, por ejemplo de proyectos de investigación, donde el presupuesto solicitado es mayor al presupuesto disponible.

2.4.1. Clasificación con restricción en el tamaño de las categorías (CSP)

Un Problema de Clasificación con Restricciones surge de situaciones de decisión formuladas en un problema de clasificación en el cual es necesario introducir

especificaciones en los tamaños de las categorías. Estas especificaciones son las restricciones sobre el tamaño de las categorías.

Los problemas CSP difieren de los problemas de clasificación usuales. Un problema de clasificación usual se refiere a una evaluación absoluta de objetos (no se requiere comparación entre objetos). Sin embargo, las restricciones sobre el tamaño de las categorías implican cierta evaluación relativa entre objetos.

Para justificar el interés de las restricciones en el tamaño de las categorías para el Análisis Multicriterio para la Toma de Decisiones, se mostrarán varios problemas de decisión ilustrativos realistas (Mousseau et al., 2003) en los que esta noción puede jugar un papel importante en el proceso de modelado.

Ejemplo 1.

Considere un gerente de crédito en una institución financiera que decide si procede o no conceder créditos a los clientes. Su papel es el de aceptar / rechazar expedientes de crédito o posiblemente referirse a su superior para los casos difíciles o ambiguos. Su decisión se basa en los diversos elementos documentados en el expediente. Este problema de decisión se puede formular a través de una segmentación tricotómica de múltiples criterios (aceptar / consultar superior / rechazar).

Sin embargo, el gerente de crédito no puede enviar muchos archivos (no más de 10 % en promedio) a su superior. En tal caso, es natural concebir un modelo de clasificación con tres categorías en las que la clase central tiene una restricción de tamaño del 10%.

Ejemplo 2.

Cada año, el director de una empresa quiere dividir un presupuesto de bonos entre sus colaboradores. Considera cuatro niveles de bonos (A: alto, B: mediano, C: pequeño, D: nada) de acuerdo a varios criterios de desempeño que se discuten en el comienzo del año con sus colaboradores (los montos de los paquetes de los bonos A, B y C también son establecidos en el inicio del año).

La política de primas del director es tal que muy pocos colaboradores reciben un bono A,

unos pocos más consiguen un bono B, una parte significativa logra conseguir un bono C y una gran parte de ellos no reciben bonificación. En esta situación, la forma de la distribución del tamaño de categoría puede ser considerado como "creciente" (o "decreciente" de acuerdo con la forma en que las categorías están numeradas) .

Ejemplo 3.

Cada año, el responsable del programa de formación de la Universidad se enfrenta al mismo problema en la definición de los cursos de lenguas extranjeras. Él quiere dividir un grupo de estudiantes (aproximadamente 100) en tres grupos de diferentes niveles (principiantes, intermedios y avanzados). La asignación de un estudiante a una clase específica se basa en sus habilidades (expresión oral, comprensión auditiva, expresión escrita, ...). Sin embargo, con el fin de ser "justo" con los profesores y estudiantes, las tres clases están destinadas a ser "no muy diferentes" de tamaño. Tal problema de decisión, obviamente, se puede formular como un problema de clasificación multicriterio. Una de las particularidades de este problema consiste en el tamaño "uniforme " de las categorías representando las tres clases.

Algunos CSP pueden formularse como problemas de ranking o selección si las categorías están ordenadas, y siempre que no haya una definición más o menos absoluta de la categoría. Si existe una restricción de tamaño que indica que la primera categoría debe contener k alternativas, esto correspondería a una selección de las mejores k alternativas de un conjunto, pero solo en caso de que no sea importante verificar que esas alternativas se apegan a la definición absoluta de la categoría correspondiente. Si existen limitaciones de tamaño que indican que las primeras k_1 alternativas pertenecen a C_1 , k_2 alternativas pertenecen a C_2 , etc., esto se puede lograr ordenando las alternativas de la mejor a la peor y romper el ranking en n_{cat} segmentos (de nuevo asumiendo que el DM no esté interesado en lograr la correspondencia de la asignación con la definición de la categoría). Hay situaciones obvias de CPS que no se pueden formular usando una formulación relativa (elección o ranking). Una situación como el ejemplo 2 no puede ser resuelta por el ranking de las alternativas, porque no sabríamos qué segmento con un 10% de las alternativas (en el ranking) se debe seleccionar.

Estos tres problemas de decisión ilustran situaciones de clasificación en los que el tamaño de la categoría interviene de alguna manera en el proceso de modelado. Un analista de ayuda a la decisión que diseñe un modelo para estas situaciones, debe tener en cuenta en el proceso de modelado la información concerniente a los tamaños de las categorías, de acuerdo a como sea requerido por el DM.

2.4.2. Definición de un problema CSP

La definición formal de los problemas de clasificación con restricción en el tamaño de las categorías fue hecha por (Mousseau et al., 2003). El tamaño de una categoría en un modelo de clasificación con restricción en el tamaño de las categorías C_k se define como la proporción por la cual un vector de evaluación asigna a un objeto a la categoría C_k . Sea $K = \{1, 2, \dots, M\}$; proponen las siguientes propiedades sobre el tamaño proporcional de la categoría k -ésima, denotado por $\mu(C_k)$:

$$\mu(C_k) \geq 0, \forall k \in K$$

$$\sum_1^M \mu(C_k) = 1$$

$$\mu(\cup_{k \in K'} C_k) = \sum_{k \in K'} \mu(C_k), \forall K' \subset K$$

En un CSP se pueden definir diferentes tipos de limitaciones sobre el tamaño de las categorías (Köksalan et. Al, 2017):

- Una categoría puede incluir un porcentaje de todos los objetos.
- El número de objetos asignados a una categoría debe ser menor que el número de objetos asignados a otra categoría.
- El número de objetos asignados a una categoría puede ser definido como un intervalo.

2.5. ELECTRE TRI-B

En esta sección se describirá el método de clasificación ELECTRE TRI-B, basado en relaciones de no inferioridad. Este método es el método insignia del enfoque europeo para resolver problemas de clasificación.

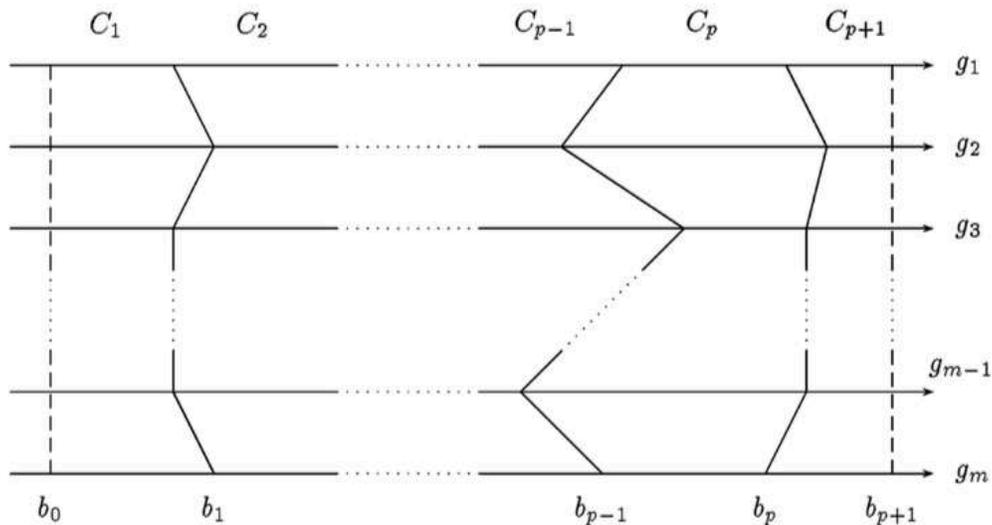


Figura 2.3. Definición de las categorías usando perfiles como límites

Fuente: (Mousseau et. al, 2000).

ELECTRE TRI-B es un método de clasificación multicriterio que asigna las n acciones de un conjunto $A = \{a_1, a_2, \dots, a_n\}$ a M categorías predefinidas. La asignación de una acción a resulta de la comparación de a con los perfiles que definen los límites de las categorías. Sea $G = \{g_1, g_2, \dots, g_m\}$ el conjunto de índices de los criterios y $B = \{b_1, b_2, \dots, b_p\}$ el conjunto de los perfiles que definen M categorías, siendo b_h el límite superior de la categoría C_h y el límite inferior de la categoría C_{h+1} , $h = 1, 2, \dots, p$ (ver la Figura 2.3, donde los perfiles b_{p+1} y b_0 corresponden a los objetos ideal y anti-ideal, respectivamente). Sin pérdida de generalidad, se considera que la preferencia se incrementa con el valor de cada criterio.

Esquemáticamente, ELECTRE TRI-B asigna objetos a categorías siguiendo dos pasos consecutivos:

- Construcción de una relación de no-inferioridad (outranking) S que caracteriza cómo las acciones se comparan con los límites de las categorías.
- Explotación de la relación S para asignar cada acción a una categoría específica.

2.5.1. Construcción de la relación de no-inferioridad S

ELECTRE TRI-B construye una relación de no-inferioridad S , para validar la aserción de aSb_h (y b_hSa), cuyo significado es " a es al menos tan buena como b_h ". Los umbrales de preferencia e indiferencia ($P_j(b_h)$ y $Q_j(b_h)$) constituyen la información preferencial intra-criterios. Estos consideran la naturaleza imprecisa de las evaluaciones $g_j(a)$. $Q_j(b_h)$ especifica la diferencia más grande $g_j(a) - g_j(b_h)$ que preserva la indiferencia entre a y b_h en el criterio g_j ; $P_j(b_h)$ representa la diferencia más pequeña $g_j(a) - g_j(b_h)$ compatible con una preferencia a favor de a en el criterio g_j (Mousseau et. al, 2000).

Dos tipos de parámetros de preferencia inter-criterio intervienen en la construcción de S :

Dos tipos de parámetros de preferencia inter-criterio intervienen en la construcción de S :

- El conjunto de los coeficientes de los pesos (w_1, w_2, \dots, w_m) es usado en la prueba de concordancia cuando se calcula la importancia relativa de las coaliciones de los criterios a favor de la aserción aSb_h .
- El conjunto de umbrales de pre-veto ($U_1(b_h), U_2(b_h), \dots, U_m(b_h)$) y veto ($V_1(b_h), V_2(b_h), \dots, V_m(b_h)$) son utilizados en la prueba de discordancia; $v_j(b_h)$ representa la diferencia más pequeña $g_j(b_h) - g_j(a)$ incompatible con la aserción aSb_h .

2.5.2. Procedimiento de construcción

ELECTRE TRI-B construye una relación de no-inferioridad S usando un índice de credibilidad $\sigma(a, b_h) \in [0,1]$ y $\sigma(b_h, a) \in [0,1]$, que representa el grado de credibilidad de la aserción de aSb_h y b_hSa , respectivamente $\forall a \in A, \forall h \in B$. La relación de no-

inferioridad en este trabajo será calculada de acuerdo con Roy (1990) y se utilizará la simplificación de Mousseau y Dias (2006), que introduce el umbral de pre-veto.

La relación de no-inferioridad proporciona un valor real entre 0 y 1 para indicar la fuerza preferencial entre el objeto a y el objeto b_h . Se considera que todos los criterios son favorables. Los símbolos Q_j, P_j, U_j, V_j ($j = 1, 2, \dots, n$) denotan los umbrales de indiferencia, preferencia, pre-veto y veto respectivamente. La evaluación de la alternativa a en el criterio g_j es definida por $g_j(a)$.

La relación de no-inferioridad es construida con los índices de concordancia global y de discordancia como se describe a continuación.

- Índice de concordancia

El índice de concordancia global es definido como sigue:

$$C(a, b_h) = \sum_{j=1}^n w_j c_j(a, b_h) \quad (2.3)$$

donde el índice de concordancia por criterio $c_j(a, b_h)$ es calculado de la manera siguiente:

$$c_j(a, b_h) = \begin{cases} 0 & g_j(b_h) - g_j(a) \geq P_j \\ 1 & g_j(b_h) - g_j(a) \leq Q_j \\ \text{de otra manera} & \frac{g_j(a) - g_j(b_h) + P_j}{P_j - Q_j} \end{cases} \quad (2.4)$$

w_j es el peso del criterio g_j , y $C(a, b_h)$ es el índice de concordancia global que soporta la aserción aSb_h , que implica que “ a es al menos tan bueno como b_h ”.

- Índice de discordancia

El índice de discordancia considera el efecto de cualquier criterio discordante en la validación de aSb_h empleando la información preferencial de los umbrales de pre-veto U y veto V . Para reflejar la capacidad otorgada por el j – esimo para rechazar la afirmación aSb_h el índice de discordancia se define como:

$$d_j(a, b_h) = \begin{cases} 0 & g_j(b_h) - g_j(a) \leq U_j \\ 1 & g_j(b_h) - g_j(a) \geq V_j \\ \text{de otra manera} & \frac{g_j(b_h) - g_j(a) - U_j}{V_j - U_j} \end{cases} \quad (2.5)$$

- Índice de credibilidad de la relación de no-inferioridad

El índice de credibilidad σ de la relación de no-inferioridad es calculado como sigue:

$$\sigma(a, b_h) = C(a, b_h) \cdot \prod_{j \in G: d_j(a, b_h) > C(a, b_h)} \frac{1 - d_j(a, b_h)}{1 - C(a, b_h)} \quad (2.6)$$

Observe que cuando $d_j(a, b_h) = 1$, el índice de credibilidad $\sigma(a, b_h) = 0$.

La definición del índice de credibilidad es basada en las siguientes premisas (Figueira et. al, 2016):

- a) Cuando no hay criterios discordantes, la credibilidad de la relación de no-inferioridad es igual al índice de concordancia global, $\sigma(a, b_h) = C(a, b_h)$.
- b) Cuando un criterio discordante activa el poder de veto, la aserción aSb_h no es creíble y el índice de credibilidad $\sigma(a, b_h) = 0$.
- c) Para las situaciones restantes en las cuales el índice de concordancia global es estrictamente más bajo que el índice de discordancia sobre el criterio discordante ($C(a, b_h) < d_j(a, b_h)$), el índice de credibilidad llega a ser más bajo que el índice de concordancia global, por el efecto de oposición en ese criterio, $\sigma(a, b_h) < C(a, b_h)$.

2.5.3. Procedimiento de explotación

Como la asignación de los objetos a las categorías no se obtiene directamente de la relación S , una fase de explotación es necesaria; esta requiere que la relación S utilice un λ - corte:

la aserción aSb_h es considerada válida si $\sigma(a, b_h) \geq \lambda$ y $(\sigma(b_h, a) \geq \lambda)$, respectivamente), siendo λ un nivel de corte tal que $\lambda \in [0.5, 1]$. Este nivel de corte determina la situación de preferencia entre a y b_h :

- $\sigma(a, b_h) \geq \lambda$ y $\sigma(b_h, a) \geq \lambda \Rightarrow aSb_h$ y $b_hSa \Rightarrow aIb_h$. a es indiferente a b_h .
- $\sigma(a, b_h) \geq \lambda$ y $\sigma(b_h, a) < \lambda \Rightarrow aSb_h$ y no $b_hSa \Rightarrow a > b_h$. a es preferido a b_h (debilmente o estrictamente).
- $\sigma(a, b_h) < \lambda$ y $\sigma(b_h, a) \geq \lambda \Rightarrow$ no aSb_h y $b_hSa \Rightarrow b_h > a$. b_h es preferido a a (debilmente o estrictamente).
- $\sigma(a, b_h) < \lambda$ y $\sigma(b_h, a) < \lambda \Rightarrow$ no aSb_h y no $b_hSa \Rightarrow aRb_h$. a es incomparable a b_h .

Observe que b_0 y b_{p+1} son definidos tal que $b_{p+1} > a$ y $aSb_0, \forall a \in A$. El rol del procedimiento de explotación es analizar la manera en la cual una acción a es comprada con los perfiles y cómo determinar la categoría a la cual a debería ser asignada. Se tienen dos procedimientos de asignación disponibles:

Procedimiento pesimista (o conjuntivo):

- a) Comparar a sucesivamente con b_i , para $i = p, p - 1, \dots, 0$
- b) Al encontrar el primer perfil b_h tal que aSb_h , asignar a a la categoría C_{h+1} ($a \rightarrow C_{h+1}$).

Procedimiento optimista (o disyuntivo):

- a) Comparar a sucesivamente con b_i , para $i = p, p - 1, \dots, 0$.
- b) Al encontrar el primer perfil b_h tal que aSb_h , asignar a a la categoría C_{h+1} ($a \rightarrow C_{h+1}$).

Si b_h y b_{h-1} denotan el perfil superior e inferior de la categoría C_h , el procedimiento pesimista asigna a a la categoría C_h más alta tal que a sobreclasifique a b_{h-1} , aSb_{h-1} . Cuando se usa este procedimiento con $\lambda = 1$, una alternativa a puede ser asignada a la categoría C_h solamente si $g_j(a)$ iguala (supera a un umbral) o excede $g_j(b_h)$ en cada criterio (regla conjuntiva) (Mousseau et. al, 2000).

El procedimiento optimista asigna a a la categoría C_h más baja para la cual el perfil inferior b_h es preferido a a , $b_{h-1} > a$. Cuando se usa este procedimiento con $\lambda = 1$, una alternativa a puede ser asignada a la categoría C_h cuando $g_j(b_h)$ excede $g_j(a)$ (en algún umbral) en al menos un criterio (regla disyuntiva) (Mousseau et. al, 2000). Cuando λ decrementa, las propiedades conjuntiva y disyuntiva de estas reglas se debilitan.

2.6. Análisis de Desagregación de Preferencias (PDA)

Muchas decisiones prácticas pueden ser modeladas usando análisis de decisión multicriterio. Los métodos multicriterio suponen a un tomador de decisiones con la capacidad de expresar sus preferencias en una estructura matemática. Por lo tanto, obtener información preferencial del DM y formalizar dicha información en un conjunto de parámetros preferenciales es un aspecto crucial en la construcción de un modelo de decisión multicriterio (Fernández et. al, 2009).

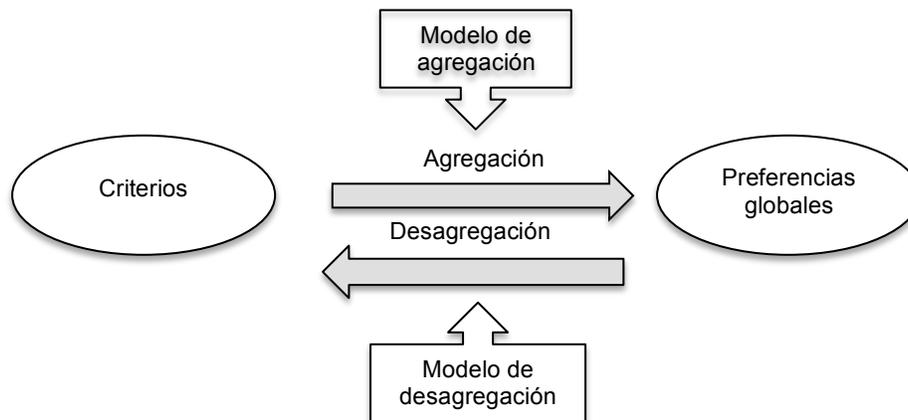


Figura 2.4. Paradigmas de agregación y desagregación en MCDA

El desarrollo de estos modelos puede ser basado en procedimientos de elicitación de parámetros de preferencia de manera directa o indirecta. En los procedimientos directos el DM debe especificar los parámetros preferenciales a través de un proceso interactivo guiado por un analista de decisión. Generalmente el DM muestra dificultad cuando se le solicita que asigne valores a los parámetros cuyo significados son poco claros para él. Por otro lado se tienen los procedimientos indirectos, que componen el paradigma del análisis

de desagregación de preferencias (PDA), el cual utiliza métodos inspirados en regresión para inferir un conjunto de parámetros preferenciales de un conjunto de ejemplos de decisión.

En el paradigma de agregación tradicional, el modelo de agregación es conocido a priori, mientras que las preferencias globales son desconocidas. Por el contrario, la filosofía de la desagregación involucra la inferencia del modelo de preferencias dada las preferencias globales, ver figura 2.4.

El análisis de desagregación provee un marco metodológico para el desarrollo de modelos de decisión usando ejemplos de decisiones tomadas por un DM (o grupo de DM's), de tal manera que el sistema de preferencia es representado en el modelo lo más preciso posible. La entrada principal usada en este proceso es un conjunto de alternativas de referencia evaluadas por el DM (ejemplos de decisión). El conjunto de referencia puede consistir de decisiones pasadas, un subconjunto de alternativas bajo consideración, o un conjunto de acciones ficticias que pueden ser fácilmente juzgadas por el DM. Dependiendo de la problemática de decisión la evaluación de las acciones de referencia puede ser expresada definiendo una estructura de orden (total, débil, parcial, etc.) o en clases clasificadas (Doumpos y Zopoundis, 2011).

El objetivo del análisis de desagregación de preferencias es inferir los parámetros preferenciales $\hat{\beta}^*$ que aproximen, con la mayor precisión posible, el sistema preferencial del DM implícito en el conjunto de acciones de referencia (de entrenamiento). Se espera que con los parámetros obtenidos, las evaluaciones realizadas con el modelo correspondiente sean consistentes con las evaluaciones que realizaría el DM. Esta metodología proporciona una base sólida para el apoyo a la decisión, siguiendo un enfoque constructivo (Doumpos y Zopoundis, 2011).

El interés de este trabajo se centra en aplicar un método PDA al método ELECTRE TRI-B para inferir un modelo preferencial completo (pesos de los criterios, umbrales, perfiles frontera y lambda) de un conjunto de entrenamiento. Dias y otros (2006) mencionan que inferir todos los parámetros simultáneamente en modelos ELECTRE, requiere de resolver un problema no lineal con restricciones no convexas, lo que es usualmente difícil.

De acuerdo con Doumpos y otros (2009), la forma relacional de estos modelos y las condiciones de veto pueden hacer imposible inferir el conjunto de parámetros preferenciales en conjuntos de datos de tamaños reales.

Los enfoques más clásicos para resolver problemas de inferencia de parámetros reducen el problema a un solo objetivo. Fernández y otros (2009) realizaron el primer trabajo que plantea la optimización multiobjetivo evolutiva para implementar un método de inferencia de parámetros para una relación de indiferencia basada en el enfoque de no-inferioridad. En su trabajo resolvieron el problema multiobjetivo mediante algoritmos genéticos.

Doumpos y otros (2009), utilizaron evolución diferencial para desarrollar un método de desagregación de preferencias utilizando el método ELECTRE TRI-B. Otras aplicaciones de la computación evolutiva para inferir parámetros de modelos basados en relaciones de no-inferioridad pueden verse en (Fernández et al., 2012; Covantes et al., 2016; Cruz et al., 2017).

2.7. Metaheurísticas

Las metaheurísticas son procedimientos de búsqueda de alto nivel que aplican ciertos tipos de reglas con base en alguna fuente de conocimiento, a fin de poder realizar una exploración más eficiente del espacio de búsqueda.

La aplicación de las técnicas metaheurísticas es especialmente interesante en caso de problemas de optimización combinatoria: problemas en las que las variables de decisión son enteras (o discretas, al menos) en las que, generalmente, el espacio de soluciones está formado por ordenaciones de valores de dichas variables. Sin embargo, las técnicas metaheurísticas se pueden aplicar también a problemas de otro tipo, como con variables continuas, por ejemplo.

La lógica de las técnicas metaheurísticas es considerar que el punto de partida es una solución (o conjunto de soluciones) que típicamente no es óptima. A partir de ella se obtienen otras parecidas, de entre las cuales se elige una que satisface algún criterio, a partir de la cual comienza de nuevo el proceso. Este proceso se detiene cuando se cumple alguna

condición establecida previamente.

Las técnicas metaheurísticas más extendidas son las siguientes: los algoritmos genéticos, algoritmos evolutivos, la búsqueda tabú, el recocido simulado, la búsqueda “scatter”, las colonias de hormigas, la técnica conocida por el nombre inglés GRASP y cúmulos de partículas PSO.

Todas las técnicas metaheurísticas tienen las siguientes características (Siarry y Michalewicz, 2008):

- Son ciegas, no saben si llegan a la solución óptima. Por lo tanto, se les debe indicar cuándo deben detenerse.
- Son algoritmos aproximativos y, por lo tanto, no garantizan la obtención de la solución óptima.
- Aceptan ocasionalmente malos movimientos (es decir, se trata de procesos de búsqueda en los que cada nueva solución no es necesariamente mejor –en términos de la función objetivo– que la inmediatamente anterior). Algunas veces aceptan, incluso, soluciones no factibles como paso intermedio para acceder a nuevas regiones no exploradas.
- Son relativamente sencillas; todo lo que se necesita es una representación adecuada del espacio de soluciones, una solución inicial (o un conjunto de ellas) y un mecanismo para explorar el campo de soluciones.
- Son generales. Prácticamente se pueden aplicar en la resolución de cualquier problema de optimización de carácter combinatorio. Sin embargo, la definición de la técnica será más o menos eficiente en la medida en que las operaciones tengan relación con el problema considerado.
- La regla de selección depende del instante del proceso y de la historia hasta ese momento. Si en dos iteraciones determinadas, la solución es la misma, la nueva solución de la siguiente iteración no tiene por qué ser necesariamente la misma. En general, no lo será.

Aunque las soluciones que ofrecen los técnicas metaheurísticas no son las óptimas y, en general, ni siquiera es posible conocer la proximidad de las soluciones al óptimo, permiten estudiar problemas de gran complejidad de una manera sencilla y obtener soluciones suficientemente buenas en tiempos razonables.

La mayoría de los algoritmos metaheurísticos son inspirados en la naturaleza, han sido desarrollados basados en alguna abstracción de la naturaleza. La naturaleza ha evolucionado por millones de años y ha encontrado soluciones perfectas para la mayoría para casi todos los problemas conocidos. Se puede aprender del éxito de la forma en que la naturaleza soluciona sus problemas y desarrollar algoritmos heurísticos y/o metaheurísticos inspirados en la naturaleza.

Los dos mayores componentes de un algoritmo metaheurísticos son: selección de la mejor solución y aleatoriedad. La selección de los mejores asegura que las soluciones converjan al óptimo, mientras la aleatoriedad evita que las soluciones se estanquen en óptimos locales e incrementa la diversidad de las soluciones. Una buena combinación de estos dos componentes usualmente asegurarían que el óptimo global sea alcanzado.

En las secciones siguientes se analizarán dos metaheurísticas: algoritmos genéticos y algoritmo evolutivo diferencial. Se revisarán sus operadores y su procedimiento básico.

2.7.1. Algoritmo genético

Un algoritmo genético está basado en una metáfora biológica, en el cual un estado sucesor es generado combinando dos estados padres. Un algoritmo genético inicia con un conjunto de k individuos generados aleatoriamente, este conjunto es llamado población inicial del algoritmo genético. Cada individuo es representado como una cadena de datos, mayormente utilizados 0's y 1's.

La idea del algoritmo genético (AG) fue propuesta por John Holland y sus colaboradores en los 60s y 70s; es un modelo o abstracción basado de evolución biológica basado en la teoría de la selección natural de Charles Darwin. Holland fue el primero en utilizar los operadores de cruce, mutación y selección en el estudio de sistemas artificiales y adaptativos. Estos operadores forman la parte esencial de la parte de un algoritmo genético

como parte de la estrategia de solución de un problema.

Muchas variantes de algoritmos genéticos han sido desarrollados y aplicados en un amplio rango de problemas de optimización, coloreo de grafos, reconocimiento de patrones, en sistemas discretos como el problema del viajero, en sistemas continuos como el diseño eficiente de fuselajes en ingeniería aeroespacial, en el mercado financiero y en optimización multiobjetivo en el área de ingeniería.

Hay muchas ventajas de los algoritmos genéticos sobre los algoritmos tradicionales de optimización, y las dos ventajas principales son: la habilidad de tratar con problemas complejos de optimización y el paralelismo. Los algoritmos genéticos pueden tratar con varios tipos de optimización si la función objetivo es estacionaria o no estacionaria (cambia en el tiempo), lineal o no lineal, continua o discontinua, o con ruido aleatorio. Los individuos de la población actúan como agentes independientes que puede explorar en espacio de búsqueda en múltiples direcciones. Esta característica hace a los algoritmos genéticos ideales para paralelizarlos en su implementación.

Sin embargo, los algoritmos genéticos también tiene algunas desventajas. La formulación de la función objetivo, el tamaño de la población inicial que se utilizará, la selección de parámetros importantes como la tasa de mutación y cruce, el criterio de selección de la nueva población deben ser cuidadosamente elegidos. Una selección inapropiada causará problemas de convergencia al algoritmo genético, o simplemente esto producirá resultados sin sentido. A pesar de estas desventajas, los algoritmos genéticos permanecen como uno de los algoritmos de optimización más ampliamente usados en optimización no lineal moderna.

2.7.1.1. Algoritmo

El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario), y usa selección probabilística.

El algoritmo básico es el siguiente (Coello, 2017):

- Generar (aleatoriamente) una población inicial.

- Calcular aptitud de cada individuo.
- Seleccionar (probabilísticamente) en base a aptitud.
- Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.
- Ciclar hasta que cierta condición se satisfaga.
- La representación tradicional es la binaria, tal y como se ejemplifica en la figura 2.5.

A la cadena binaria se le llama “cromosoma”. A cada posición de la cadena se le denomina “gene” y al valor dentro de esta posición se le llama “alelo”.

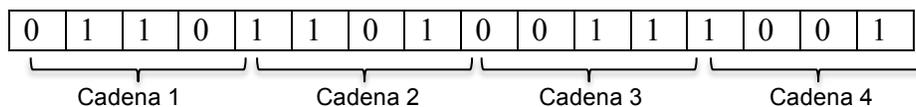


Figura 2.5. Ejemplo de la codificación (mediante cadenas binarias) usada tradicionalmente con los algoritmos genéticos

Para poder aplicar el algoritmo genético se requiere de los 5 componentes básicos siguientes:

1. Una representación de las soluciones potenciales del problema.
2. Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
3. Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su “aptitud”.
4. Operadores genéticos (selección, cruce y mutación) que alteren la composición de los hijos que se producirán para las siguientes generaciones.
5. Valores para los diferentes parámetros que utiliza el algoritmo genético (tamaño de la población, probabilidad de cruce, probabilidad de mutación, número máximo de generaciones, etc.) .

2.7.1.2. Selección

Una parte fundamental del funcionamiento de un algoritmo genético es, sin lugar a dudas, el proceso de selección de candidatos a reproducirse. En el algoritmo genético este proceso de selección suele realizarse de forma probabilística (es decir, aún los individuos menos aptos tienen una cierta oportunidad de sobrevivir), a diferencia de las estrategias evolutivas, en las que la selección es extintiva (los menos aptos tienen cero probabilidades de sobrevivir).

Las técnicas de selección usadas en algoritmos genéticos pueden clasificarse en tres grandes grupos:

- Selección proporcional: ruleta, sobrante estocástico, universal estocástica, muestreo determinístico, escalamiento sigma,, selección por jerarquías, selección de Boltzmann.
- Selección mediante torneo.
- Selección de estado uniforme.

En este trabajo se emplearán la selección por ruleta y la selección por torneo.

2.7.1.2.1. Selección por ruleta

Esta técnica fue propuesta por De Jong (1975), y ha sido el método más comúnmente usado desde los orígenes de los algoritmos genéticos. A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente la población está ordenada en base al ajuste por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0..1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

El algoritmo es simple, pero ineficiente (su complejidad es $O(n^2)$). Asimismo, presenta el

problema de que el individuo menos apto puede ser seleccionado más de una vez. Sin embargo, buena parte de su popularidad se debe no sólo a su simplicidad, sino al hecho de que su implementación se incluye en el libro clásico sobre AGs de David Goldberg (1989).

El algoritmo de la Ruleta (de acuerdo a De Jong, 1975) es el siguiente:

- Calcular la suma de los valores de las aptitudes de los individuos para calcular \bar{f}

$$\bar{f} = \frac{\sum_{i=1}^n f(ind_i)}{n}$$
 donde $f(ind_i)$ es la aptitud del individuo i y n es el número de individuos en la población.
- Calcular los valores esperados $Ve_i = \frac{f_i}{\bar{f}}$
- Calcular la suma de valores esperados T
- Repetir n veces (n es el tamaño de la población):
 - Generar un número aleatorio $r, r \in [0.0, T]$
 - Ciclar a través de los individuos de la población sumando los valores esperados Ve hasta que $\sum Ve \geq r$.
 - El individuo que haga que esta suma exceda el valor de r es el seleccionado.

El ejemplo siguiente muestra el funcionamiento de esta técnica de selección.

Individuo	Aptitud	Ve
(1)	25	0.35
(2)	81	1.13
(3)	36	0.51
(4)	144	2.01
	$\sum = 286$	$\sum = 4.00$

$$\bar{f} = \frac{286}{4} = 71.5 \quad T = 4.0 \quad r \in [0.0, 4.0]$$

Se genera un número aleatorio, por ejemplo $r = 1.3$

Se busca al individuo cuya suma de valores $V_e \geq r$

Ind 1 suma = $0.35 < r$

Ind 2 suma = $1.48 > r$

El individuo 2 es seleccionado. Esto se repite hasta tener el tamaño de selección deseado.

Una desventaja de esta técnica es que el peor individuo puede ser seleccionado varias veces y su complejidad es $O(n^2)$.

2.7.1.2.2. Selección por torneo

La selección mediante torneo estudiada en la tesis doctoral de Brindle (1981) y fue propuesta por Wetzel (1983).

La idea básica del método es seleccionar con base en comparaciones directas de los individuos.

El algoritmo de la versión determinística es el siguiente:

- Barajar los individuos de la población.
- Escoger un número p de individuos (típicamente 2, para un torneo binario).
- Compararlos con base en su aptitud.
- El ganador del “torneo” es el individuo más apto.
- Debe barajarse la población hasta seleccionar N padres (donde N es el tamaño de la población).

Algunas ventajas de esta técnica son:

- Su baja complejidad, se requieren “ n ” competencias de este tipo para completar una generación, por lo tanto, el algoritmo es $O(1)$.
- La técnica es eficiente y fácil de implementar.
- No requiere escalamiento de la función de aptitud (usa comparaciones directas).

Una desventaja es que puede introducir una presión de selección muy alta, a los individuos menos aptos no se les da oportunidad de sobrevivir.

Un ejemplo de su funcionamiento, es el siguiente (Coello, 2017):

Individuo	Aptitud	Barajar	Ganadores
(1)	254	(2) (6)	(6)
(2)	47	(4) (1)	(1)
(3)	457	(1) (3)	(3)
(4)	194	(6) (5)	(6)
(5)	85	(5) (4)	(4)
(6)	310	(2) (3)	(3)

Padres:

(6) y (1), (3) y (6), (4) y (3)

2.7.1.3. Cruce

En los sistemas biológicos, la cruce es un proceso complejo que ocurre entre parejas de cromosomas. Estos cromosomas se alinean, luego se fraccionan en ciertas partes y posteriormente intercambian fragmentos entre sí.

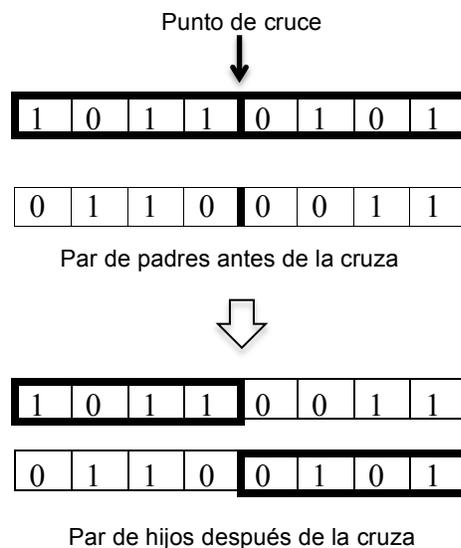


Figura 2.6. Cruce de un punto aleatorio

En la figura 2.6 se ilustra el proceso de cruce de un punto, este esquema de cruce será utilizado en este trabajo.

2.7.1.4. Mutación

La mutación se considera como un operador secundario en los algoritmos genéticos canónicos. Es decir, su uso es menos frecuente que el de la cruce.

En la práctica, se suelen recomendar porcentajes de mutación de entre 0.001 y 0.01 para la representación binaria.

Algunos investigadores, sin embargo, han sugerido que el usar porcentajes altos de mutación al inicio de la búsqueda, y luego decrementarlos exponencialmente, favorece el desempeño de un AG (Terence 1989).

Otros autores sugieren que $pm = \frac{1}{L}$ (donde L es la longitud de la cadena cromosómica) es un límite inferior para el porcentaje óptimo de mutación (Bäck 1993).

El papel que juega la mutación en el proceso evolutivo, así como su comparación con la cruce, sigue siendo tema frecuente de investigación y debate en la comunidad de computación evolutiva. En este capítulo estudiaremos diferentes técnicas de mutación que se han propuesto en la literatura especializada.

2.7.1.5. Elitismo

Se denomina elitismo al mecanismo utilizado en algoritmos genéticos para asegurar que los cromosomas de los miembros más aptos de una población se pasen a la siguiente generación sin ser alterados por ningún operador genético. Ha sido demostrado que el AG requiere de elitismo (o sea, retener intacto al mejor individuo de cada generación) para poder converger al óptimo (Gunter, 1994).

Usar elitismo asegura que la aptitud máxima de la población nunca se reducirá de una generación a la siguiente. Sin embargo, no necesariamente mejora la posibilidad de localizar el óptimo global de una función. No obstante, es importante hacer notar que se ha

demostrado que el uso de elitismo es vital para poder demostrar convergencia de un algoritmo genético (Coello, 2018).

2.7.1.6. Algoritmos genéticos vs otras técnicas evolutivas

- El AG usa selección probabilística al igual que la Programación Evolutiva, y en contraposición a la selección determinística de las Estrategias Evolutivas.
- El AG usa representación binaria para codificar las soluciones a un problema, por lo cual se evoluciona el genotipo y no el fenotipo como en la Programación Evolutiva o las Estrategias Evolutivas.
- El operador principal en el AG es la cruce, y la mutación es un operador secundario. En la Programación Evolutiva, no hay cruce y en las Estrategias Evolutivas es un operador secundario.
- Ha sido demostrado que el AG requiere de elitismo (o sea, retener intacto al mejor individuo de cada generación) para poder converger al óptimo.

2.7.1.7. Aplicaciones

Algunas aplicaciones de los AGs son las siguientes (Goldberg,1989):

- Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- Aprendizaje de máquina (sistemas clasificadores)
- Bases de datos (optimización de consultas)
- Reconocimiento de patrones (por ejemplo, imágenes)
- Generación de gramáticas (regulares, libres de contexto, etc.)
- Planeación de movimientos de robots
- Predicción

2.7.2. Algoritmo evolutivo diferencial

El algoritmo evolutivo diferencial, mejor conocido como evolución diferencial (DE, por sus

siglas en inglés “Differential Evolution”), fue propuesto por Prince and Storn (1997). Este enfoque heurístico fue desarrollado para resolver problemas no lineales o con funciones no diferenciables. Al igual que un algoritmo genético tradicional, el algoritmo DE implementa operadores genéticos de mutación, cruce y selección sobre una población inicial hasta que se cumpla el criterio de terminación establecido. La elección de utilizar DE en este trabajo se debe a su buen desempeño en problemas de optimización continuos, principalmente con respecto a clasificación y clustering (Doumpos et. al, 2009).

2.7.2.1. Algoritmo

Asumiendo una generación G de P soluciones en \mathbb{R}^N , la metodología de la estrategia DE puede ser descrita como sigue:

- **Mutación:** por cada solución g_i^G de la generación actual ($i = 1, \dots, P$) una solución mutada es generada g_i^{mut} . Diferentes esquemas de mutación han sido propuestos en Storn (1996). En este trabajo será usado el esquema DE/rand-to-best/1, el cual a dado buenos resultados dentro del contexto específico del problema (Doumpos et. al, 2009). En este esquema, la solución mutada es obtenida como sigue:

$$g_i^{mut} = g_i^G + m_c(g_{best}^G - g_i^G) + m_c(g_h^G - g_s^G)$$

donde $m_c > 0$ es la constante de mutación, g_{best}^G es el mejor individuo de la generación actual (él de mayor desempeño, para problema de maximización) y h, s ($h \neq s$) son índices seleccionados aleatoriamente del conjunto $\{1, \dots, P\}$. Si el desempeño más alto es alcanzado por múltiples soluciones en una misma generación, g_{best}^G es igual al primero de estos (el de índice menor).

- **Cruce:** Este operador tiene la finalidad de incrementar la diversidad de la población. El operador de cruce es realizado utilizando el esquema exponencial (Prince et al., 2005) como se muestra en la figura 2.7. Inicialmente es seleccionado un índice aleatorio del vector de soluciones $\ell \in \{1, \dots, N\}$ y números aleatorios $\beta_{\ell+1}, \dots, \beta_N$ son generados con una distribución uniforme $[0,1]$. La solución cruzada

g_i^{cr} es definida como sigue:

$$g_i^{cr} = \begin{cases} g_{ij}^{mut} & \text{si } \ell \leq j \leq j^* - 1 \\ g_{ij}^G & \text{de otra manera} \end{cases} \quad (2.7)$$

donde g_{ij}^{mut} es el j th elemento de la solución mutada g_i^{mut} y j^* es el primer índice tal que $\beta_j > CR$, con $CR \in [0,1]$ siendo CR una probabilidad de cruce definida por el usuario. Si no hay un índice j^* tal que $\beta_j > CR$ ($\beta_{\ell+1}, \dots, \beta_N \leq CR$), entonces

$$g_i^{cr} = (g_{i1}^G, g_{i2}^G, \dots, g_{i,\ell-1}^G, g_{i,\ell}^{mut}, g_{i,\ell+1}^{mut}, \dots, g_{iN}^{mut}) \quad (2.8)$$

para valores altos de CR ($CR \approx 1$) el esquema de cruce exponencial es similar al cruce de un punto.

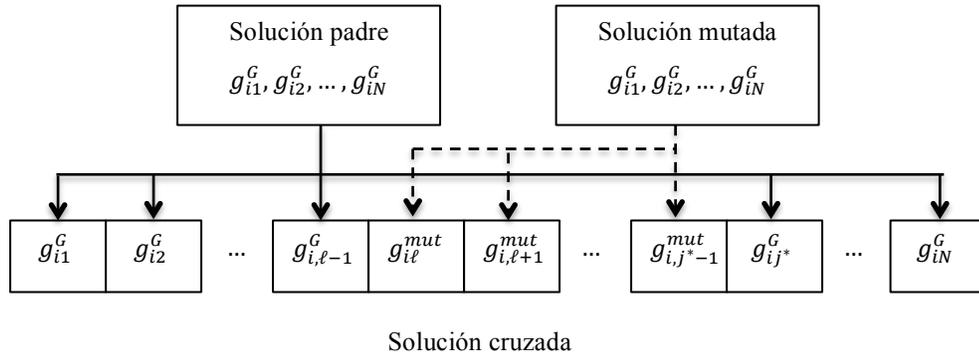


Figura 2.7. Operador de cruce exponencial del algoritmo DE

- **Selección:** después de las operaciones de mutación y cruce, una etapa de selección es realizada. Para una solución padre dada g_i^G de la generación actual y la solución cruzada asociada g_i^{cr} , el i th miembro de la generación siguiente es especificado como sigue:

$$g_i^{G+1} = \begin{cases} g_i^{cr} & \text{si } f(g_i^{cr}) > f(g_i^G) \\ g_i^G & \text{de otra manera} \end{cases} \quad (2.9)$$

donde f es la función objetivo para un problema de maximización.

Estos operadores de mutación, cruce y selección se estarán realizando hasta que se cumpla el criterio de terminación establecido.

2.7.2.2. Aplicaciones

Algunas aplicaciones de los algoritmos DE son las siguientes (Prince and Storn, 2005):

- Optimización (industrial, numérica, etc.)
- Procesamiento de imágenes
- Diseño de filtros digitales

2.7.3. Algoritmo Evolutivo Multiobjetivo basado en Descomposición (MOEA/D)

El Algoritmo Evolutivo Basado en Descomposición (MOEA/D)(Zhang et al.,2007) explícitamente descompone el problema multiobjetivo (MOP) en N subproblemas de optimización. Resuelve simultáneamente estos subproblemas considerando una población de soluciones. En cada generación, la población es compuesta de la mejor solución encontrada hasta el momento por cada subproblema. Las relaciones de vecindad entre los subproblemas se basan en las distancias entre sus coeficientes de agregación. Cada problema en MOEA/D es optimizado utilizando solamente información de sus subproblemas vecinos. MOEA/D tiene las siguientes características:

1. Provee una manera simple y eficiente de introducir un enfoque de descomposición en computación evolutiva multiobjetivo.
2. Debido a que MOEA/D optimiza N subproblemas de optimización en lugar de resolver el MOP como uno solo, cuestiones como la asignación de desempeño y conservación de diversidad que causan dificultades para algoritmos MOEAs de no descomposición, pueden ser más sencillas de manejar en el marco de MOEA/D.

3. MOEA/D tiene una complejidad computacional más baja en cada generación que NSGA-II y otros conocidos algoritmos multiobjetivo.

2.7.3.1. Algoritmo

MOEA/D necesita descomponer el MOP en consideración. Cualquier enfoque de descomposición puede servir para este propósito; en esta implementación se utilizará el enfoque de Tchebycheff propuesto en Zhang et al., 2007. En este enfoque el problema de optimización escalar es de la forma

$$\text{minimizar } g^{te}(x|\rho, z^*) = \max_{1 \leq i \leq num_obj} \{\rho_i |f_i(x) - z_i^*|\} \quad (2.10)$$

sujeto a $x \in \Omega$, donde $z^* = (z_1^*, \dots, z_{num_obj}^*)$ es el punto de referencia, $z_i^* = \{f_i(x)|x \in \Omega\}^3$ para cada $i = 1, \dots, num_obj$.

Sea ρ_1, \dots, ρ_N un conjunto de vectores de pesos separados uniformemente y sea z^* el punto de referencia. El problema de aproximación del frente de Pareto puede ser descompuesto en N subproblemas escalares usando el enfoque de Tchebycheff, y la función objetivo del j-ésimo subproblema es

$$g^{te}(x|\rho^j, z^*) = \max_{1 \leq i \leq num_obj} \{\rho_i^j |f_i(x) - z_i^*|\} \quad (2.10)$$

donde $\rho^j = (\rho_1^j, \dots, \rho_{num_obj}^j)$. MOEA/D minimiza todas estas N funciones objetivo simultáneamente en una sola corrida.

En la generación t, MOEA/D mantiene:

- Una población de N puntos $x^1, \dots, x^N \in \Omega$, donde x^i es la solución actual del i-ésimo subproblema.
- FV^1, \dots, FV^N , donde FV^i es el desempeño de x^i , $FV^i = F(x^i)$ para cada $i = 1, \dots, N$.
- $z = (z_1, \dots, z_{num_obj})$, donde z, es el mejor valor encontrado hasta el momento para el objetivo f_i .

- Una población externa EP, la cual es utilizada para almacenar soluciones no dominadas encontradas durante la búsqueda.

El algoritmo trabaja como sigue:

Entradas:

- MOP
- Un criterio de parada.
- N , número de subproblemas considerados en MOEA/D.
- N vectores de pesos separados uniformemente: ρ_1, \dots, ρ_N .
- T_{vecinos} , el número de vectores de pesos en el vecindario para cada vector de pesos.

Salida:

- EP, población externa.

El algoritmo de MOEA/D se describe a continuación:

Paso1- Inicialización:

- 1.1 $EP = \emptyset$
- 1.2 Calcular las distancias Euclidianas entre todos los vectores de pesos y encontrar los T_{vecinos} de cada vector de pesos. Para cada $i = 1, \dots, N$, calcular $D(i) = \{i_1, \dots, i_{T_{\text{vecinos}}}\}$, donde $\rho^{i_1}, \dots, \rho^{i_{T_{\text{vecinos}}}}$ son los T_{vecinos} vectores de pesos más cercanos a ρ^i .
- 1.3 Generar aleatoriamente una población inicial x^1, \dots, x^N . Hacer $FV^i = F(x^i)$.
- 1.4 Inicializar $z = (z_1, \dots, z_{\text{num_obj}})$, de acuerdo al problema específico a resolver.

Paso 2- Actualización:

For $i = 1, \dots, N$

- 2.1 Reproducción: seleccionar aleatoriamente dos índices k, l de $D(i)$ y formar una nueva solución y de x^k y x^l usando operadores genéticos.

- 2.2 Mejora: Aplicar un método de reparación a y de acuerdo al problema específico a resolver para producir y' .
- 2.3 Actualización de z : por cada $j = 1, \dots, num_obj$, si $z_j > f_j(y')$, entonces $z_j = f_j(y')$.
- 2.4 Actualización del vecindario de soluciones: por cada índice $j \in D(i)$, si $g^{te}(y'|\rho^j, z) \leq g^{te}(x^j|\rho^j, z)$, entonces $x^j = y'$ y $FV^j = F(y')$.
- 2.5 Actualizar EP:
- Remove de EP todos los vectores dominados por $F(y')$.
- Agregar $F(y')$ a EP si no hay vectores en EP dominados por $F(y')$.

Paso 3 – Criterio de parada:

Si el criterio de parada se cumple entonces se detiene y EP es la salida; de lo contrario se va al paso 2.

Capítulo 3.

Implementación del Análisis de Desagregación de Preferencias PDA

En esta sección se muestra la manera en que se implementó el método PDA para la desagregación de preferencias del conjunto T de entrenamiento. Para ello se utilizaron dos metaheurísticas, algoritmos genéticos y evolución diferencial. Finalmente se muestran los resultados comparativos de la implementación del método PDA empleando un algoritmo genético y utilizando evolución diferencial.

3.1. Parámetros a extraer por el método PDA

Como se describió en la sección 2.7, el objetivo del análisis de desagregación de preferencias es inferir los parámetros preferenciales que aproximen, lo más precisamente posible (ver figura 3.1), el sistema preferencial del DM implícito en el conjunto de acciones de referencia (de entrenamiento).



Figura. 3.1- Método de análisis de desagregación de preferencias

El método PDA implementado en este trabajo extrae del conjunto de referencia T los parámetros preferenciales siguientes:

- Pesos de los criterios, w_i
- Umbral de indiferencia, Q_i
- Umbral de preferencia, P_i
- Umbral de pre-veto, U_i
- Umbral de veto, V_i
- Perfiles frontera, $b_{j,i}$
- Umbral de credibilidad, λ

$j = 1, 2, \dots, M - 1$. Representa el índice de la frontera

$i = 1, 2, \dots, m$. Representa el número de criterio a que corresponde el parámetro

La metodología propuesta infiere del conjunto de entrenamiento un modelo preferencial completo (Q, P, U, V, W , perfiles frontera y λ) que permite una mejor caracterización de un DM. El modelo tiene la ventaja de representar de una manera más completa a un DM que los procedimientos implementados en la literatura para problemas CSP (vea Tabla 1.2), que, o bien utilizan modelos compensatorios con información cardinal, o no toman en consideración los efectos de veto.

La inferencia de parámetros preferenciales para este trabajo está restringida a la implementación de un procedimiento de desagregación de preferencias con el método ELECTRE TRI-B. Cuando se requiere inferir todos los parámetros preferenciales simultáneamente en los modelos basados en métodos ELECTRE, es necesario resolver un problema no lineal con restricciones no convexas, lo cual es usualmente difícil (Mosseau and Slowinski, 1998; Dias et al., 2002).

Los algoritmos evolutivos han mostrado ser una herramienta poderosa para resolver problemas difíciles en varios campos de la ciencia, en particular para tratar la no linealidad y optimización global en tiempo polinomial (Coello et al., 2007).

En las siguientes secciones se mostrarán las implementaciones del método PDA empleando un algoritmo genético (sección 3.2) y mediante la utilización de evolución diferencial (sección 3.3).

3.2. Implementación del método PDA mediante un algoritmo genético

El problema de la desagregación de preferencias que se requiere implementar implica un problema no lineal, ya que se busca obtener la información sobre la capacidad de veto en las preferencias del decisor implícitas en el conjunto de entrenamiento. Para ello se utilizará un algoritmo genético; el algoritmo genético es una herramienta que ha dado buenos resultados en este tipo de problemas, muestra de ello son los trabajos de Fernández et al., 2009, Fernández et al., 2012, Fernández et al., 2014 y Covantes et al., 2016. En estos

trabajos se emplean algoritmos genéticos para realizar un procedimiento de desagregación de preferencias sobre un conjunto de acciones de referencia (conjunto de entrenamiento).

El algoritmo genético es una metaheurística evolutiva que se basa en el principio de la evolución de Darwin cuyos operadores principales son: selección, cruce y mutación. El algoritmo genético implementado se muestra en la figura 3.2. La efectividad del algoritmo genético depende en gran medida de los valores asignados a los parámetros que requiere. Para esta implementación los valores de los parámetros utilizados son:

- Población de 50 individuos
- El tamaño de selección es de 20 individuos
- Probabilidad de cruce de 80%
- Probabilidad de Mutación de 0.1%
- Se emplea elitismo

En la implementación se utilizan dos poblaciones de 50 individuos: la población de trabajo y una población externa. La población de trabajo es la población inicial que es generada aleatoriamente, sobre la cual se realizan las operaciones de los operadores de cruce y mutación, y las técnicas de selección y reemplazo utilizadas. La población externa es una población que es utilizada para guardar los mejores individuos que encuentre el algoritmo genético en su búsqueda. La población externa no está sujeta a los operadores de cruce y mutación, ni a la técnica de selección. Solo se le aplica una técnica de reemplazo cuando la población está llena y se encuentra un individuo nuevo, que es mejor al peor que contiene hasta ese momento. El nuevo individuo ingresa y sale el peor individuo que contiene la población externa. Esta población externa es llamada población semilla, y es la población que será pasada a la siguiente etapa de la implementación del algoritmo multiobjetivo evolutivo MOEA/D (sección 4.4).

Los pasos que realiza el algoritmo genético son:

1. Generar población inicial de 50 individuos
2. Evalúa desempeño de la población
3. Hace la selección por ruleta de 20 individuos

4. Aplica el cruce y la mutación con la probabilidad configurada
5. Evalúa el desempeño de la selección
6. Se aplica una técnica de reemplazo por ruleta a los 70 individuos que se tienen en este punto (50 de la población inicial y 20 de la selección) para extraer 20 individuos. Los 50 individuos que quedan forman la nueva población.
7. Se asegura elitismo
8. Actualiza la población semilla
9. Si cumple un criterio de parada (vea criterios de parada) termina, si no regresa al punto 3

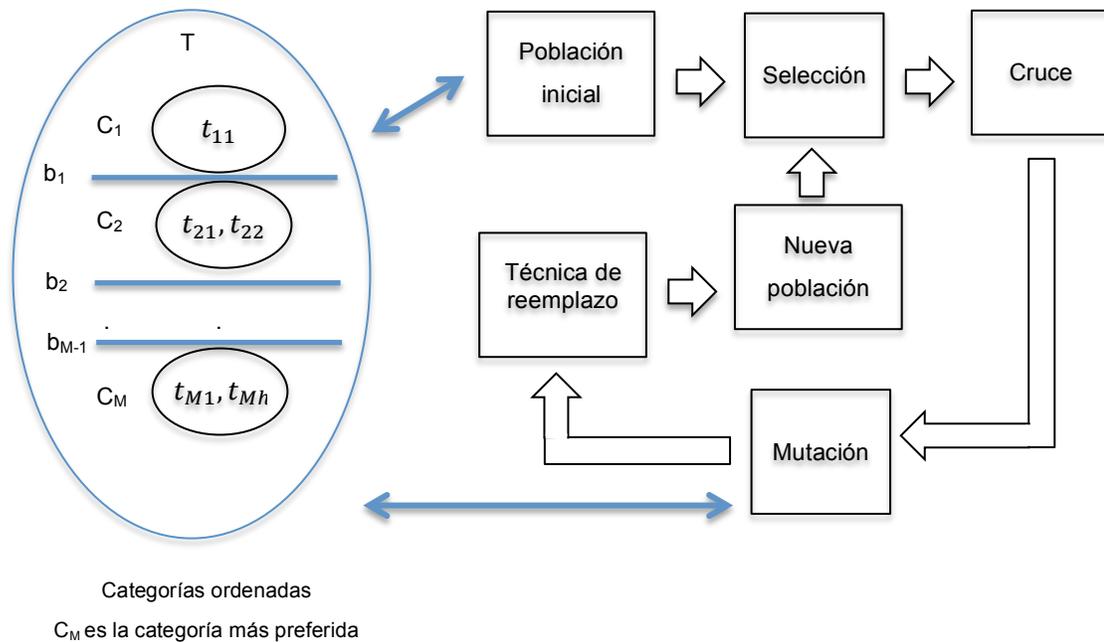


Figura 3.2 - Algoritmo genético implementado

El algoritmo genético implementado resuelve el problema mono-objetivo siguiente:

$$\underset{t \in T}{\text{maximizar}} F(t) = OC(t) \quad (3.1)$$

donde OC es el número de objetos bien clasificados del conjunto T.

El algoritmo genético resuelve el problema (3.1), buscando conjuntos de parámetros preferenciales (Q, P, U, V, W , perfiles frontera y λ) que logren clasificar a la totalidad de los objetos del conjunto T de manera idéntica al decisor.

Inicialmente se genera una población de 50 individuos, la cual es comparada con el conjunto de referencia para encontrar su función de aptitud (fitness). El desempeño se evalúa con OC, que representa el número de objetos del conjunto de entrenamiento clasificados correctamente y, OD1, el número de objetos clasificados una categoría arriba o debajo de la categoría correcta. Se considera que un individuo tiene mejor desempeño cuanto mayor es el valor de OC. El valor de OD1 en el desempeño de un individuo es utilizado para desempate, en caso de que dos o más individuos tenga el mismo valor de OC.

La población inicial cumple las restricciones siguientes:

- $0 < Q_i < P_i < U_i < V_i < g_i(max)$
- $\sum_{i=1}^m w_i = 1$ (6)
- $g_i(min) < b_{1,1} < b_{2,2} < \dots < b_{M-1,m} < g_i(max)$
- $0.66 \leq \lambda \leq 0.8$

$g_i(max)$, $g_i(min)$ denotan los desempeños máximo y mínimo de cada criterio.

Cada individuo de la población del algoritmo genético contiene un conjunto de parámetros preferenciales (P , Q , U , V , pesos de los criterios, perfiles frontera y λ) que representan las preferencias de un decisor. La codificación de un individuo se muestra en la figura 3.3. En las primeras posiciones del arreglo se encuentran los valores de los umbrales Q , P , U y V por cada criterio; luego los valores de los pesos por criterio; enseguida están ubicados los perfiles frontera que corresponden a cada criterio en cada categoría y finalmente el umbral de credibilidad. Con esta codificación se tienen individuos con longitud, puntos de cruce y puntos de mutación definidos por:

$$longitud\ del\ individuo = (m*5) + m*(M-1) + 1 \quad (3.2)$$

$$puntos\ de\ cruce = 2m + 1 \quad (3.3)$$

$$puntos\ de\ mutación = longitud\ del\ individuo - 1 \quad (3.4)$$

donde m es el número de criterios y M es el número de categorías.

La figura 3.3 también muestra los $2m+1$ puntos de cruce posibles para cada individuo. Los puntos de cruce se hacen a nivel de criterio, con esto cuando se aplica el operador de cruce

se asegura que se mantiene la consistencia entre criterios sin la necesidad de hacer pruebas de consistencia y reparación. Esto con la finalidad de reducir el esfuerzo computacional que estas operaciones requieren.

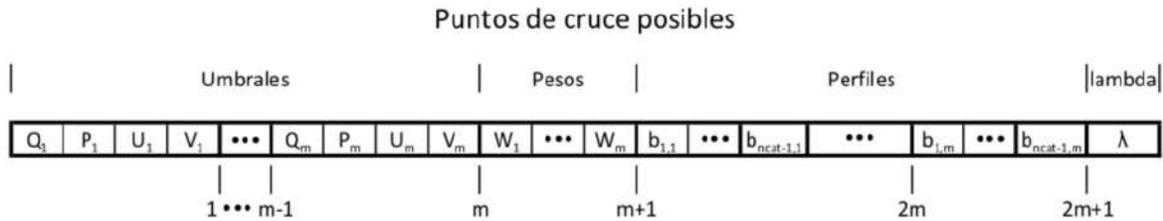


Figura 3.3 - Codificación y puntos de cruce de un individuo

A diferencia del operador de cruce que se aplica a nivel de criterio, el operador de mutación puede ser aplicado a cada posición del individuo. Por esta razón se tienen *longitud del individuo - 1* puntos de mutación. Para realizar la operación de mutación se requiere implementar operaciones de reparación para mantener la consistencia entre criterios. En la tabla 3.1 se muestran ejemplos de la longitud, número de puntos de cruce y número de puntos de mutación para individuos de 3, 4 y 5 categorías con 3, 5 y 7 criterios.

Tabla 3.1 – Ejemplos de longitud, puntos de cruce y puntos de mutación para individuos

Número de categorías	Número de criterios	Longitud del individuo	Puntos de cruce	Puntos de mutación
3	3	22	7	21
	5	36	11	35
	7	50	15	49
4	3	25	7	24
	5	41	11	40
	7	57	15	56
5	3	28	7	27
	5	46	11	45
	7	64	15	63

3.2.1. Generación de una población semilla con el algoritmo genético

El algoritmo genético empleado para implementar el método de PDA en este trabajo además de obtener los parámetros preferenciales (P , Q , U , V , pesos de los criterios, perfiles frontera y λ) implícitos en el conjunto de referencia T , genera una población semilla (población externa) con los 50 mejores individuos distintos que encontró durante su ejecución. Se define como mejor individuo al conjunto de parámetros que obtiene el método PDA que puede representar mejor las preferencias del decisor (clasifica correctamente al mayor número de objetos del conjunto de referencia), incluidos los individuos que pueden clasificar a la totalidad de objetos del conjunto de referencia T , que serán llamados individuo solución. La población semilla es distinta a la población de trabajo del algoritmo genético que se actualiza sistemáticamente durante la ejecución del procedimiento de inferencia de parámetros.

La población semilla en cualquier momento de la ejecución del método PDA contiene los mejores individuos que ha podido encontrar hasta ese momento. Para garantizar que la población tenga los mejores 50 individuos se implementaron tres criterios de parada para el método de PDA:

1. Cuando la población semilla contiene 50 individuos solución
2. Cuando el método PDA no fue capaz de encontrar 50 individuos solución, se lleva a cabo un procedimiento de “sacudidas”. Si después de esperar 100,000 iteraciones sin que se haya encontrado un nuevo individuo para la población semilla, la “sacudida” consiste en incrementar la probabilidad de mutación a 30% durante 1000 iteraciones y posteriormente restaurarla a 0.1%. Si esta secuencia se repite 3 veces seguidas se pasa al tercer criterio de parada. Si durante el procedimiento de sacudidas se introduce un nuevo individuo a la población semilla, la secuencia de sacudidas inicia de nuevo.
3. Después de 20 reinicios. Se considera un reinicio cuando no se cumplieron los dos primeros criterios de parada. En cada reinicio se genera una nueva población inicial y se va actualizando la población semilla.

Con esto se logra tener una población semilla que contenga los mejores individuos que el procedimiento de inferencia de parámetros pudo encontrar; esta población semilla será utilizada en la población inicial que se genere para el algoritmo MOEA/D del capítulo 4.

3.2.2. Experimentación del algoritmo genético con diferente técnica de selección y reemplazo

Una parte fundamental del funcionamiento de un algoritmo genético es, sin lugar a dudas, el proceso de selección de candidatos a reproducirse. En el algoritmo genético este proceso de selección suele realizarse de forma probabilística (es decir, aún los individuos menos aptos tienen una cierta oportunidad de sobrevivir), a diferencia de las estrategias evolutivas, en las que la selección es extintiva (los menos aptos tienen cero probabilidades de sobrevivir).

Las técnicas de selección usadas en algoritmos genéticos pueden clasificarse en tres grandes grupo (Coello, 2018):

- Selección proporcional
- Selección mediante torneo
- Selección de estado uniforme

En este trabajo se utilizarán las técnicas de selección por torneo y selección por ruleta. La selección por ruleta es una técnica de selección proporcional, en la que los individuos mejores (con mayor aptitud) son los que tienen mayores posibilidades de ser elegidos.

La técnica de selección por ruleta fue propuesta por DeJong (1975), es posiblemente el método más utilizado desde los orígenes de los Algoritmos Genéticos (Blickle y Thiele, 1995). Es un método muy sencillo, pero ineficiente a medida que aumenta el tamaño de la población. Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez (para mas detalle vea sección 2.8.1.2.1).

En la selección por torneo la idea principal consiste en realizar la selección en base a comparaciones directas entre individuos. Existen dos versiones de selección mediante torneo:

- Determinística
- Probabilística

La versión determinística será empleada en este trabajo, en ella se selecciona al azar un número p de individuos (se utilizó $p=2$, torneo binario). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del algoritmo genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Para determinar qué técnica de selección sería la mas adecuada para este trabajo se realizaron experimentos con tres configuraciones distintas:

1. Selección y reemplazo por torneo
2. Selección por torneo y reemplazo por ruleta
3. Selección y reemplazo por ruleta

Las tablas 3.2 y 3.3 muestran las tres versiones de algoritmos genéticos que se implementaron y probaron con diferentes configuraciones para selección y reemplazo. Los resultados que se muestran corresponden a los 400 experimentos realizados por cada una de las configuraciones, variando la probabilidad de mutación en 1%, 0.5%, 0.3% y 0.1%. Lo que se midió fue el número de iteraciones que requirió el algoritmo genético para encontrar un conjunto de parámetros de preferencia (Q , P , U , V , W , perfiles frontera y λ) que clasificaba correctamente a todos los elementos del conjunto de referencia. Los valores máximo y mínimo corresponden al número máximo y mínimo de iteraciones que se tuvo en

los 400 experimentos para una configuración dada. Para los 400 experimentos el método PDA implementado logro clasificar a la totalidad de los objetos del conjunto de referencia.

En el anexo 1 se muestran el conjunto de entrenamiento utilizado para los experimentos y su clasificación correcta. El conjunto de entrenamiento consta de 28 objetos evaluados en 10 criterios y clasificados en 3 categorías.

Inicialmente se planteó un esquema de experimentación que consiste en generar una población inicial diferente para cada uno de los 400 experimentos de cada configuración y de cada probabilidad de mutación. Esto hace que el resultado del algoritmo genético sea muy sensible a los datos de entrada . Los resultados de este esquema de experimentación se muestran en la tabla 3.2. Las celdas de color verde contienen al los valores menores que se obtuvieron para la media y la desviación estándar en el número de iteraciones que requirió cada configuración probada hasta clasificar los 28 objetos del conjunto de entrenamiento. Estos resultados no permiten ser muy concluyente para decidir cuál de las 3 configuraciones probadas es preferible.

Tabla 3.2. Resultados de los experimentos para las tres configuraciones implementadas

		Medidas (iteraciones)	Probabilidad de mutación			
			1%	0.50%	0.30%	0.10%
1ra configuración	Selección: Torneo Reemplazo: Torneo	Media	53,714.32	100,260.51	198,109.70	459,915.46
		Desviación estándar	113,058.84	193,882.66	394,781.12	909,745.48
		Mínimo	2	10	7	0
		Máximo	874,537	2,230,534	3,896,181	7,138,067
2da configuración	Selección: Torneo Reemplazo: Ruleta	Media	58,299.33	102,494.51	192,865.79	372,364.60
		Desviación estándar	112,667.05	210,437.08	433,885.37	746,587.08
		Mínimo	8	11	1	13
		Máximo	1,042,448	2,041,115	3,956,156	5,888,381
3ra configuración	Selección: Ruleta Reemplazo: Ruleta	Media	51,082.08	91,308.24	173,751.42	413,399.35
		Desviación estándar	105,339.83	162,395.19	278,866.79	758,382.26
		Mínimo	18	5	7	12
		Máximo	839,781	1,467,721	3,265,345	5,836,092

Para mejorar este inconveniente se modificó el esquema de experimentación, se generará una misma población inicial y se buscará la solución con cada una de las tres configuraciones. Con esto se tendrá un mismo punto de inicio del algoritmo genético para cada una de las tres configuraciones y permitirá medir con mayor certeza la eficiencia de cada configuración.

El objetivo de cada una de las tres configuraciones es encontrar un conjunto de parámetros preferenciales (Q, P, U, V, W , perfiles frontera y λ) que logren asignar de manera idéntica al decisor los objetos del conjunto T de entrenamiento. Las tres configuraciones logran clasificar correctamente a la totalidad de los objetos en todas las 400 pruebas hechas para cada una. Con estos experimentos se pretende definir cual de las tres configuraciones requiere de menos iteraciones para encontrar la solución, para decidir cual de las tres configuraciones se va a utilizar en la implementación del método PDA.

Tabla 3.3. Resultados de los experimentos con la misma población inicial

		Medidas (iteraciones)	Probabilidad de mutación			
			1%	0.50%	0.30%	0.10%
1ra configuración	Selección: Torneo Reemplazo: Torneo	Media	65,980.53	140,268.92	167,304.57	531,507.00
		Desviación estándar	139,119.53	252,931.44	404,694.99	996,730.86
		Mínimo	8	7	6	10
		Máximo	1,274,159	1,919,752	5,829,422	8,801,938
2da configuración	Selección: Torneo Reemplazo: Ruleta	Media	61,013.10	121,954.44	162,811.38	526,088.67
		Desviación estándar	117,301.99	237,739.34	323,545.72	1,078,700.14
		Mínimo	12	7	9	10
		Máximo	1,147,689	2,070,312	4,167,863	8,074,423
3ra configuración	Selección: Ruleta Reemplazo: Ruleta	Media	49,454.57	100,836.34	157,599.95	509,561.58
		Desviación estándar	119,413.64	213,341.41	315,713.64	1,027,438.43
		Mínimo	8	7	6	8
		Máximo	1,549,393	1,999,831	4,485,121	8,724,590

La tabla 3.3 muestra los resultados obtenidos al considerar la misma población inicial para las tres configuraciones. La 3ra configuración de selección y reemplazo por ruleta, es la que obtiene la media del número de iteraciones más baja para los cuatro valores de mutación probados.

Se aplicó la prueba no paramétrica de Mann-Whitney con un grado de confianza de 95% a los resultados obtenidos en la tabla 3.3. El resultado de esta prueba estadística asegura que la 3ra configuración tiene una media menor que la 1ra configuración para valores de mutación de 1%, 0.5% y 0.1%; y menor que la 2da configuración cuando se utilizó el operador de mutación con un valor de 1% y 0.5%. Mientras que en el resto de valores de la media comparados de las otras configuraciones (mutación de 0.3% para la 1ra configuración, y 0.3% y 0.1% para la segunda configuración) no resultaron menores que el obtenido por la 3ra configuración. Por ello, se utilizará la 3ra configuración (selección y reemplazo por ruleta) para la implementación del método PDA. En el anexo 4 se muestran las gráficas de la prueba de normalización a los datos y los resultados de la prueba no paramétrica de Mann-Whitney realizada.

El algoritmo genético obtiene un conjunto de soluciones que representan cada una un conjunto de parámetros preferenciales (umbrales, pesos de los criterios, perfiles frontera y umbral de indiferencia). Estas soluciones se clasifican en dos tipos: soluciones perfectas y soluciones imperfectas. Una solución es perfecta cuando el conjunto de parámetros puede clasificar de manera correcta a la totalidad de los objetos de referencia, de lo contrario es considerada una solución imperfecta.

El método PDA implementado solo considera los objetos de referencia y no considera las restricciones al tamaño de las categorías, lo que implica que aún cuando sea una solución perfecta no necesariamente cumplirá las restricciones impuestas por el DM a los tamaños en las categorías.

3.3. Evolución diferencial

En esta sección se implementará el método PDA mediante un algoritmo genético que utiliza evolución diferencial. Este enfoque metaheurístico fue desarrollado para resolver

problemas no lineales o con funciones no diferenciables. Al igual que un algoritmo genético tradicional, el algoritmo DE implementa operadores genéticos de mutación, cruce y selección sobre una población inicial hasta que se cumpla el criterio de terminación establecido. Además se mostrará el modelado matemático de un esquema de transformación para los parámetros preferenciales.

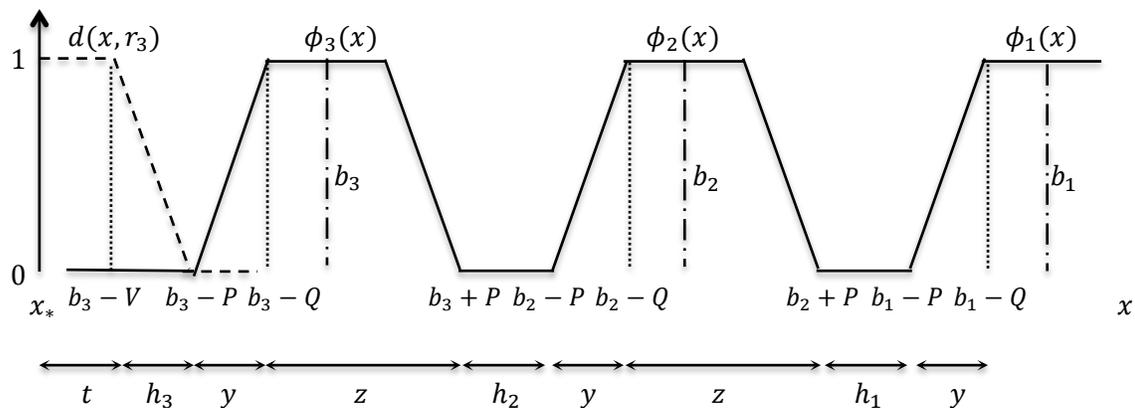


Figura 3.4. Ilustración del esquema de representación (cf. Doumpos et. Al 2009)

3.3.1. Esquema de transformación

Doumpos et. al., (2009) mencionan que la representación directa de los parámetros de preferencia (umbrales, pesos de los criterios, perfiles frontera y umbral de credibilidad) no permite una búsqueda eficiente en el espacio de solución. Para mejorar este inconveniente proponen un esquema de representación como el mostrado en la figura 3.4. Esta figura da una ilustración gráfica del esquema de representación propuesto y el rol de las variables t_j, y_j, z_j, h_{ij} en un caso de cuatro categorías ($k = 4$). Por simplicidad los índices (i y j) no se han puesto en la figura.

Este esquema de representación tiene las transformaciones siguientes:

Umbrales

$$Q_j = \frac{z_j - y_j}{2} \quad (3.5)$$

$$P_j = \frac{z_j + y_j}{2} \quad (3.6)$$

$$V_j = b_{k-1} - x_{j^*} + P_j \quad (3.7)$$

Perfiles frontera

$$b_{ij} = x_{j^*} + t_j + \sum_{\ell=i}^{k-1} h_{\ell j} + \frac{2^{(k-i)-1}}{2} (y_j + z_j) \quad (3.8)$$

$i = 1, \dots, k - 1$. Representa el número de perfil.

$j = 1, 2, \dots, m$. Representa el número de criterio a que corresponde el parámetro.

Estas ecuaciones están sujetas a las restricciones:

$$t_j + \sum_{i=1}^{k-1} h_{ij} + (k - 1)(y_j + z_j) \leq x_j^* - x_{j^*} \quad (3.9)$$

$$z \geq y \quad (3.10)$$

$$t, y, z, e, H \geq 0 \quad (3.12)$$

$$e \in \{0,1\} \quad (3.13)$$

Donde:

$$x_j^* = \max_{i \in X^*} \{x_{ij}\} \quad (3.14)$$

$$x_{j^*} = \min_{i \in X^*} \{x_{ij}\} \quad (3.15)$$

t, y, z, e son los vectores de las variables de decisión t_j, y_j, z_j, e_j ($j = 1, \dots, m$).

H es la matriz de las variables de decisión h_{ij} ($i = 1, \dots, k - 1$)($j = 1, \dots, m$).

e es el vector de la variable de decisión e_j para indicar si un criterio posee veto ($e_j = 1$) o no ($e_j = 0$).

Algunas de las ventajas del esquema de representación son que asegura que la separabilidad entre las fronteras de las categorías y tiene implícito que $P > Q$.

El esquema de transformación propuesto por Doumpos et al., 2009 no puede ser aplicada directamente en el problema de este trabajo, ya que no modela el umbral de pre-veto. Para resolver este inconveniente se realizó el modelado matemático del esquema de transformación propuesto por Doumpos et. al., 2009 para considerar el umbral de pre-veto. La figura 3.5 muestra el esquema de transformación que considera ese el umbral y con ello

el conjunto preferencial completo (umbrales, pesos de los criterios, perfiles frontera y umbral de credibilidad). La figura 3.5 da una ilustración gráfica del esquema de representación propuesto y el rol de las variables $s_j, t_j, y_j, z_j, h_{ij}$ en un caso de cuatro categorías ($k = 4$). Por simplicidad los índices (i y j) no se han puesto en la figura.

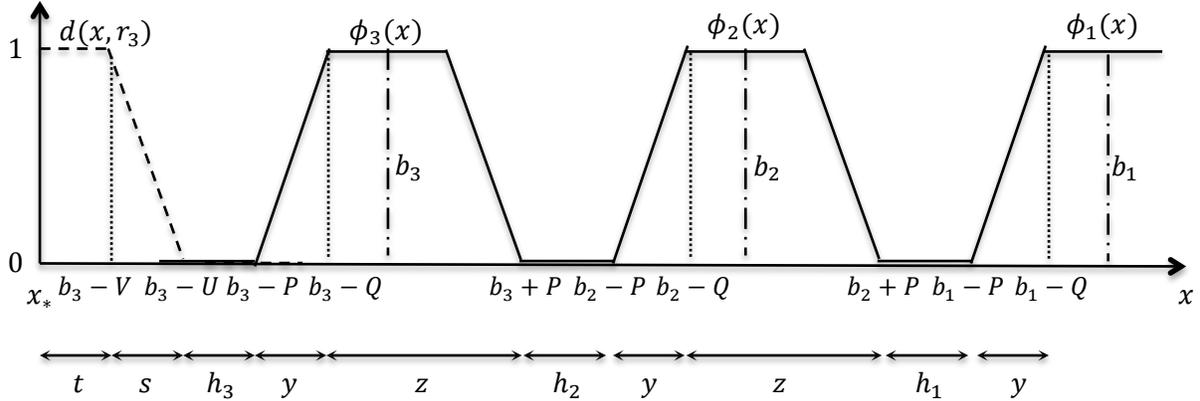


Figura 3.5. Ilustración del esquema de representación modificado

El esquema de representación tiene las transformaciones siguientes:

Umbrales

$$Q_j = \frac{z_j - y_j}{2} \quad (3.16)$$

$$P_j = \frac{z_j + y_j}{2} \quad (3.17)$$

$$U_j = b_{k-1} + P_j \quad (3.18)$$

$$V_j = b_{k-1} + s_j + P_j \quad (3.19)$$

Perfiles frontera

$$b_{ij} = x_{j*} + s_j + t_j + \sum_{\ell=i}^{k-1} h_{\ell j} + \frac{2^{(k-i)} - 1}{2} (y_j + z_j) \quad (3.20)$$

$i = 1, \dots, k - 1$. Representa el número de perfil.

$j = 1, 2, \dots, m$. Representa el número de criterio a que corresponde el parámetro.

Estas ecuaciones están sujetas a las restricciones:

$$s_j + t_j + \sum_{i=1}^{k-1} h_{ij} + (k-1)(y_j + z_j) \leq x_j^* - x_{j*} \quad (3.21)$$

$$z \geq y \quad (3.22)$$

$$s, t, y, z, e, H \geq 0 \quad (3.23)$$

$$e \in \{0,1\} \quad (3.24)$$

Donde:

$$x_j^* = \max_{i \in X^*} \{x_{ij}\} \quad (3.25)$$

$$x_{j*} = \min_{i \in X^*} \{x_{ij}\} \quad (3.26)$$

s, t, y, z, e son los vectores de las variables de decisión s_j, t_j, y_j, z_j, e_j ($j = 1, \dots, m$).

H es la matriz de las variables de decisión h_{ij} ($i = 1, \dots, k-1$) ($j = 1, \dots, m$).

e es el vector de la variable de decisión e_j para indicar si un criterio posee veto ($e_j = 1$) o no ($e_j = 0$).

Este esquema de representación asegura la separabilidad entre las fronteras de las categorías y tiene implícito que $Q < P < U < V$.

3.3.2. Implementación del método PDA mediante un algoritmo evolutivo diferencial

Rainer Storm y Kenneth Price (1996) desarrollaron una rama de la computación evolutiva denominada evolución diferencial y la aplicaron para la optimización de espacios continuos. Un algoritmo de Evolución Diferencial resuelve de manera muy eficiente problemas no lineales, no diferenciables y multimodales. Este modelo evolutivo enfatiza la mutación, utiliza un operador de cruce/recombinación a posteriori de la mutación. Es una técnica no determinista basada en la evolución de una población de vectores (individuos) de valores reales que representan las soluciones en el espacio de búsqueda. La generación de nuevos individuos se lleva a cabo mediante operadores diferenciales de mutación y cruce. La generación inicial se generará de forma aleatoria y los padres son tres individuos escogidos al azar entre la población inicial. Si el valor que obtenemos de combinarlos es

mejor que el inicial nos quedamos con la mejora, en caso contrario mantenemos la opción anterior (padre principal). La figura 3.6 muestra un esquema general del algoritmo evolutivo diferencial.

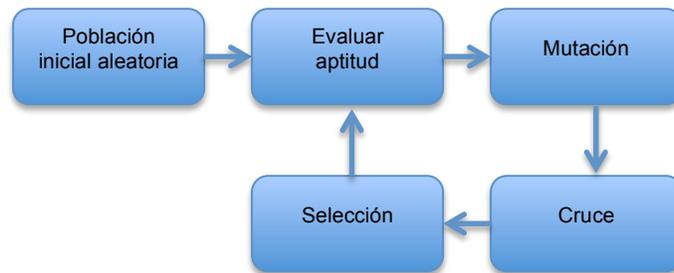


Figura 3.6. Algoritmo evolutivo diferencial

Los operadores de cruce, mutación y selección del algoritmo evolutivo diferencial que se muestran en la figura 3.6 son distintos a los de un algoritmo genético. Para más detalle de la función de cada uno de ellos vea la sección 2.7.2.

El algoritmo evolutivo diferencial implementado se muestra en el diagrama a bloques de la figura 3.7. Al igual que la implementación del método PDA con un algoritmo genético, esta implementación utiliza dos poblaciones: población de trabajo y la población externa (población semilla). Las funciones de cada población son las mismas en ambas implementaciones.

Los parámetros utilizados son:

- Población de 50 individuos.
- Probabilidad de cruce de 0.8.
- Constante de Mutación de 0.6.

Los pasos que realiza el algoritmo evolutivo diferencial son:

1. Generar población inicial de 50 individuos
2. Evalúa desempeño de la población
3. Realiza la operación de mutación
4. Aplica la operación de cruce

5. Realiza la selección
6. Si todos los objetos están clasificados correctamente termina, si no regresa al punto 3

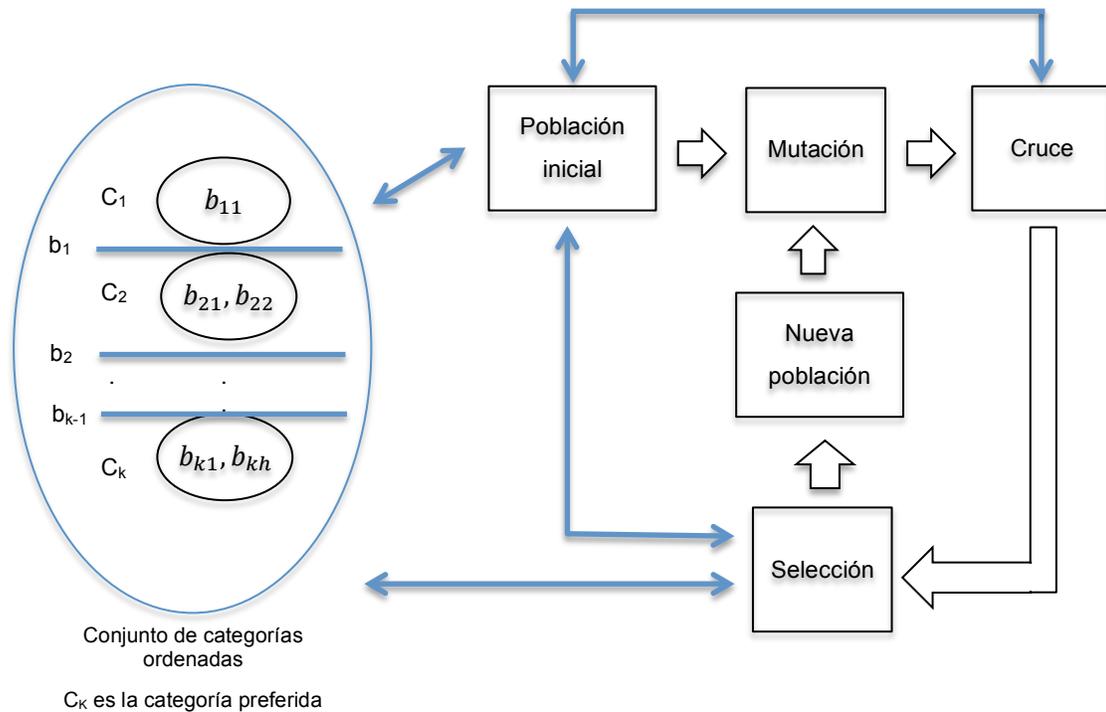


Figura 3.7. Diagrama a bloques del algoritmo evolutivo diferencial implementado

La codificación de cada individuo es mostrada en la figura 3.8, las primeras posiciones contienen a las variables s, t, y, z por criterio, luego los pesos también ubicados por criterio. Los perfiles están ubicados por categoría y por criterio, por último la variable e por criterio y λ .

La población inicial cumple las restricciones siguientes:

- $0 < Q_i < P_i < U_i < V_i < g_i(max)$
- $\sum_{i=1}^m w_i = 1$.
- $g_i(min) < b_{1,1} < b_{2,2} < \dots < b_{m,k-1} < g_i(max)$
- $0.6 \leq \lambda \leq 0.8$

$g_i(min)$, $g_i(max)$ son los desempeños máximo y mínimo de un objeto en cada criterio.

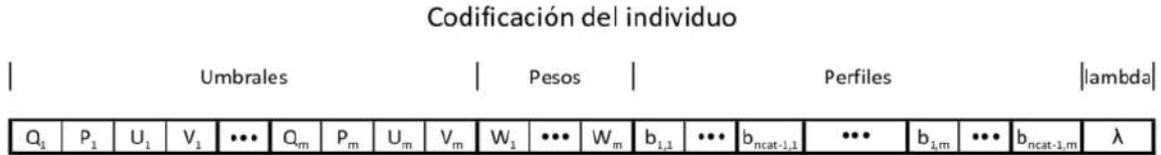


Figura 3.8. Codificación del individuo en el algoritmo evolutivo diferencial, utilizando el esquema de representación descrito en las sección anterior

El algoritmo realiza la búsqueda de las variables s, t, y, z, e, H pero al momento de evaluar la aptitud de cada individuo se requieren hacer las transformaciones indicadas en las ecuaciones 3.16 - 3.24 de la sección anterior para encontrar Q, P, U, V, B .

El algoritmo evolutivo diferencial resuelve el problema mono-objetivo siguiente:

$$\begin{aligned} \text{maximizar } F(t) = OC(t) \\ t \in T \end{aligned} \quad (3.27)$$

donde OC es el número de objetos bien clasificados del conjunto T.

Inicialmente se genera una población de 50 individuos, la cual es comparada con el conjunto de referencia para encontrar su función de aptitud (fitness). El desempeño se evalúa con OC, representa el número de objetos clasificados correctamente (del conjunto de referencia) y, OD1, el número de objetos clasificados una categoría arriba o debajo de la categoría correcta. Se considera que un individuo tiene mejor desempeño cuanto mayor es el valor de OC. El valor de OD1 en el desempeño de un individuo es utilizado para desempate en caso de que dos o más individuos tengan el mismo valor de OC.

3.3.3. Generación de una población semilla

Cada individuo de la población del algoritmo evolutivo diferencial representa un conjunto de variables s, t, y, z, e, H que están asociadas a un conjunto preferencial (P, Q, U, V, perfiles frontera y λ) por medio del esquema de transformación descrito en la sección 3.3.1. El objetivo del método PDA implementado con el algoritmo evolutivo diferencial es obtener una población semilla con los 50 mejores individuos distintos que encontró durante su ejecución. Se define como mejor individuo al conjunto de parámetros obtenido por el método PDA que puede representar mejor las preferencias del decisor, incluidos los individuos que pueden clasificar a la totalidad de objetos del conjunto de referencia T, que serán llamados individuo solución.

La población semilla en cualquier momento de la ejecución del método PDA contiene los mejores individuos que ha podido encontrar hasta ese momento. Para garantizar que la población tenga los mejores 50 individuos se implementaron tres criterios de parada para el método PDA:

1. Cuando la población semilla contiene 50 individuos solución.
2. Cuando el método PDA no fue capaz de encontrar 50 individuos solución, se lleva a cabo un procedimiento de “sacudidas”. Este procedimiento consiste en esperar 100,000 iteraciones sin que se halla encontrado un nuevo individuo para la población semilla, en ese momento se incrementa la constante de mutación a 0.8 y la probabilidad de cruce a 0.9 durante 1000 iteraciones y posteriormente regresan a 0.6 y 0.8, respectivamente. Si esta secuencia se repite 3 veces seguidas se detiene el algoritmo. Si durante el procedimiento de sacudidas se introduce un nuevo individuo a la población semilla, la secuencia de sacudidas inicia de nuevo.
3. Después de 20 reinicios. Se considera un reinicio cuando no se cumplieron los dos primeros criterios de parada. En cada reinicio se genera una nueva población inicial y se actualiza la población semilla.

Con esto se logra tener una población semilla que contenga los mejores individuos que el procedimiento de inferencia de parámetros pudo encontrar; esta población semilla será utilizada en la población inicial que se genere para el algoritmo MOEA/D del capítulo 4.

3.3.4. Experimentación

Para la experimentación se consideraron tres casos: utilizar el algoritmo genético implementado en la sección 3.2, aplicar el esquema de transformación al algoritmo genético implementado en la sección 3.2 y el algoritmo evolutivo diferencial con el esquema de transformación. El esquema de transformación utilizado es el mostrado en la sección 3.3.1.

Los experimentos se realizaron a 50 conjuntos de referencia generados artificialmente con 50, 100 y 200 objetos. Cada conjunto de referencia consideraba una evaluación en 7

critérios y 3 categorías. A cada conjunto de referencia se le aplicó 20 veces cada algoritmo del método PDA para descartar el efecto de la casualidad. Para generar los conjuntos de referencia artificiales se diseñó un generador de objetos, al que se le proporcionan como datos de entrada una combinación de parámetros (número de categorías, número de criterios y número de objetos por categoría de conjunto T). Con esto, el generador proporciona una instancia como dato de salida. La figura 3.9 muestra una ilustración del generador de instancias diseñado.

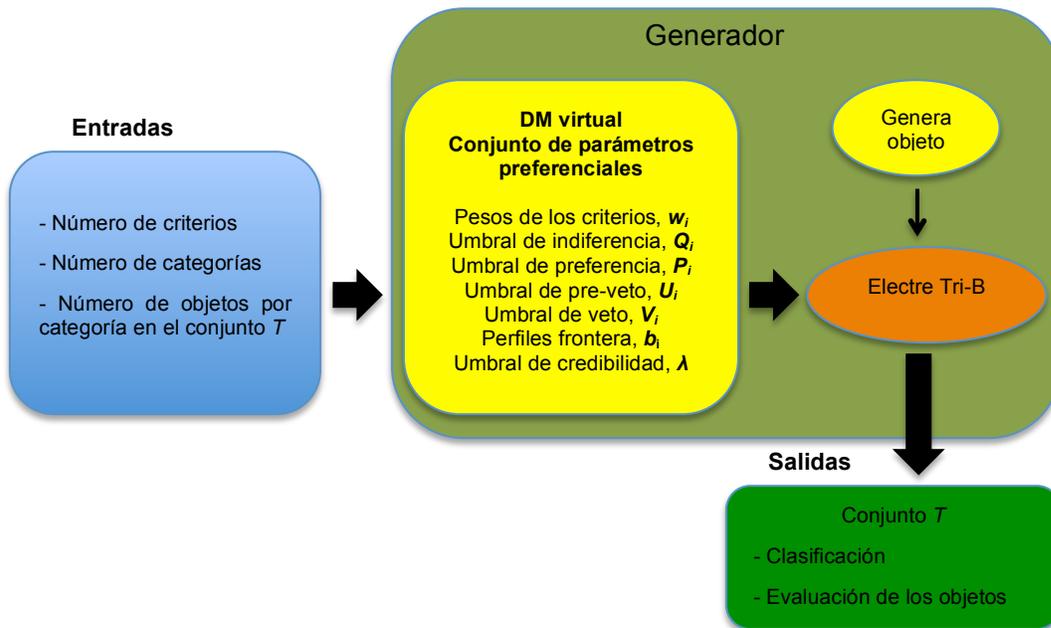


Figura 3.9. Generador de instancias para experimentación del método PDA

Cada vez que el generador forma una instancia, primero crea un modelo de un decisor (umbrales, pesos de los criterios, perfiles frontera y λ) y posteriormente genera objetos utilizando este modelo hasta llenar el conjunto T con los objetos requeridos. Antes de asignar un objeto al conjunto de referencia, el generador se asegura que no haya asignado otro objeto igual. De esta manera se asegura que para la instancia creada existe al menos una solución ideal, que es el modelo preferencial (DM virtual) que creó el generador. La instancia que proporciona el generador como salida contiene la clasificación de los objetos y sus evaluaciones en cada criterio.

3.4. Comparación de resultados de los dos métodos PDA implementados

En la tabla 3.4 se muestran la media y la desviación estándar que se obtuvieron del algoritmo genético utilizando las variables preferenciales Q, P, U, V, pesos de los criterios, perfiles frontera y umbral de credibilidad; del algoritmo genético y el algoritmo evolutivo diferencial utilizando las variables s, t, y, z, e, H del esquema de transformación.

Tabla 3.4. Comparativa de los resultados de los experimentos

	Medida	Número de objetos en el conjunto de referencia		
		50	100	200
Algoritmo Genético	Media	1.000	0.997	0.995
	Desviación estándar	0.000	0.006	0.007
Algoritmo Genético con transformaciones	Media	0.970	0.955	0.943
	Desviación estándar	0.027	0.027	0.025
Algoritmo Evolutivo Diferencial con transformaciones	Media	1.000	0.996	0.993
	Desviación estándar	0.000	0.008	0.008

Estos experimentos se realizaron a 50 conjuntos de referencia creados con el generador con 50, 100 y 200 objetos, considerando una evaluación en 7 criterios y 3 categorías. A cada conjunto de referencia se le aplicó 20 veces cada algoritmo del método PDA para descartar el efecto del azar. Las celdas de color verde indican cual de las tres implementaciones tuvo la mayor eficiencia para la inferencia de parámetros preferenciales, cuando fueron aplicadas a conjuntos de referencia de 50, 100 y 200 objetos.

El esquema de transformación propuesto por Doumpos et al., 2009 empeora la efectividad del algoritmo genético como se puede ver en la tabla 3.4. El esquema de transformación funciona muy bien en la implementación del método PDA con el algoritmo evolutivo diferencial, pero su efectividad no supera a la obtenida por al método PDA implementado con el algoritmo genético. De acuerdo con la prueba no paramétrica de Mann-Whitney

realizada con una confianza de 90%, la efectividad el algoritmo genético implementado supera al algoritmo evolutivo diferencial. Por ello el método PDA que se utilizará será el implementado con el algoritmo genético desarrollado en la sección 3.2. Los resultados de la prueba realizada se pueden encontrar en el anexo 5 al final de este trabajo.

Capítulo 4.

Enfoque multiobjetivo propuesto para el problema de clasificación ordinal multicriterio con restricciones sobre los tamaños de las categorías

La idea principal del modelo propuesto es resolver el problema de clasificación ordinal multicriterio con restricciones sobre el tamaño de las categorías empleando optimización multiobjetivo, y utilizando el método de clasificación ELECTRE TRI-B, basado en relaciones de no-inferioridad que utiliza perfiles como frontera entre las categorías.

Mediante la modelación del problema de clasificación multicriterio como un problema de optimización multiobjetivo se pretende dar flexibilidad al decisor con las restricciones preferenciales implícitas en el conjunto de referencia. Con la modelación multiobjetivo se consideran las preferencias implícitas en el conjunto de referencia y las restricciones en el tamaño de las categorías (que es también información preferencial) de manera simultánea. Los objetivos del problema tratarán, por una parte, de satisfacer la clasificación de los objetos del conjunto de referencia, y por otra, cumplir con las restricciones en el tamaño de las categorías. Con esto, el DM tendrá posibilidad de aprender sobre el problema, y la posibilidad de reconocer posibles inconsistencias en el conjunto de entrenamiento.

En esta sección se muestra la manera en que se implementó el algoritmo MOEA/D para resolver el problema de clasificación multicriterio con restricciones en el tamaño de las categorías, como un problema multiobjetivo. Además, se explica como se diseñó un generador para formar las instancias que se utilizaron en la experimentación. Finalmente, se muestran los resultados obtenidos por el algoritmo multiobjetivo MOEA/D, sin utilizar el método PDA y utilizando el método PDA.

4.1. Formulación del problema CSP como un problema de optimización multiobjetivo

Al abordar el problema de clasificación multicriterio como un problema de optimización multiobjetivo se pretende que el decisor aprenda sobre el problema, y tenga la posibilidad de reconocer posibles inconsistencias en el conjunto de entrenamiento. De tal manera que dentro de las soluciones obtenidas puede haber opciones que asignen algunos objetos de referencia de manera distinta al conjunto de entrenamiento, permitiendo con esto, cumplir mejor con las restricciones a los tamaños de las categorías.

La propuesta para resolver el problema CSP utiliza el algoritmo multiobjetivo basado en descomposición MOEA/D. El algoritmo multiobjetivo toma como entradas: i) la información preferencial implícita en el conjunto de referencia; ii) las restricciones en los tamaños de las categorías impuestas por el decisor; e iii) el conjunto de objetos a clasificar (conjunto A). El método de clasificación ordinal utilizado con el algoritmo multiobjetivo es ELECTRE TRI-B.

El problema multiobjetivo a minimizar es el siguiente:

$$\begin{aligned} \text{minimizar } F(a, t) = & (V_T(t), V_{C_1}(\mathbf{a}), V_{C_2}(\mathbf{a}), \dots, V_{C_M}(\mathbf{a})) \\ & t \in T \\ & \mathbf{a} \in A \end{aligned} \quad (4.1)$$

donde :

V_T representa las inconsistencias relativas al conjunto de entrenamiento T . Se considera una inconsistencia cuando un objeto de referencia es clasificado en una categoría que no le corresponde.

$V_{C_1}, V_{C_2}, \dots, V_M$ representan las violaciones a las restricciones de cada una de las M categorías cuando se clasifican los objetos del conjunto A . Se considera una violación por cada objeto del conjunto A , que falta o excede para tener el total de la restricción impuesta al tamaño de una categoría.

El número de objetivos para esta formulación es dependiente del número de categorías (M) que considere el problema a resolver, obteniéndose un problema con $M+1$ objetivos. Para

un problema con 3 categorías se tendrán 4 objetivos, para 4 categorías el problema a resolver será de 5 objetivos, etc.

En las secciones 4.2.1 y 4.2.2 se detallarán dos propuestas distintas acerca de utilizar o no un método PDA con el algoritmo multiobjetivo. En la sección 4.2.1 se explica la propuesta de emplear el algoritmo MOEA/D sin utilizar el método PDA. Es decir, el algoritmo MOEA/D recibe como entrada los objetos del conjunto de referencia, las restricciones en los tamaños de las categorías impuestas por el decisor, y el conjunto de objetos a clasificar.

La propuesta de utilizar el método PDA implementado en la sección 3.2 con el algoritmo MOEA/D, se muestra en la sección 4.2.2. Esta propuesta aplica el método PDA al conjunto de referencia, sin revisar las preferencias al tamaño de las categorías. De tal manera, que los modelos preferenciales obtenidos por el método PDA están ajustados solamente a las preferencias implícitas en el conjunto de referencia. En este supuesto, el algoritmo multiobjetivo MOEA/D utilizará la información preferencial obtenida por el método PDA, las restricciones en los tamaños de las categorías impuestas por el decisor y el conjunto de objetos a clasificar (conjunto A).

La implementación de estas dos propuestas permitirá conocer si el método PDA es de ayuda a la modelación multiobjetivo planteada, para resolver el problema de optimización multiobjetivo.

4.2. Implementación del algoritmo multiobjetivo MOEA/D

El algoritmo evolutivo MOEA/D implementado resuelve el problema de optimización (4.1). Busca la mejor solución posible tratando de tener 0 violaciones en cada uno de los $M+1$ objetivos ($V_T, V_{C_1}, V_{C_2}, \dots, V_{C_M}$). Los parámetros utilizados para el algoritmo son:

Entradas:

- MOP, problema (4.1)
- Un criterio de parada

- N , número de subproblemas considerados; 451 para 4 objetivos (3 categorías), 600 para 5 objetivos (4 categorías) y 800 para 6 objetivos (5 categorías)
- $T_{vecinos} = 20$, el número de vectores de pesos en el vecindario para cada vector de pesos
- Cruce y mutación de un punto
- Probabilidad de cruce de 80%
- Probabilidad de mutación de 0.1%

Los criterios de parada utilizados en el algoritmo MOEA/D son similares a los del método PDA, pero en este caso es la población externa (EP) la que se actualiza. Los tres criterios de parada para MOEA/D son:

1. Cuando se encuentra una solución al problema (4.1) con 0 violaciones en todos los objetivos ($V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$).
2. Cuando el algoritmo multiobjetivo MOEA/D no es capaz de encontrar un individuo solución ($V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$) se lleva a cabo un procedimiento de “sacudidas”. Este procedimiento consiste en incrementar la probabilidad de mutación a 30% durante 500 iteraciones, después de realizar 5000 iteraciones. Si esta secuencia se repite 3 veces seguidas se pasa al criterio de parada 3. Si durante el procedimiento de sacudidas se encuentra un individuo solución, el algoritmo termina.
3. Después de 20 reinicios. Se considera un reinicio cuando no se cumplieron los dos primeros criterios de parada. En cada reinicio se genera una nueva población inicial y se mantiene la población EP.

El algoritmo MOEA/D genera como salida por cada individuo solución en el frente de Pareto un conjunto de parámetros preferenciales y la clasificación de los objetos del conjunto A , atendiendo las restricciones impuestas por el DM. La población externa contiene los individuos del frente de Pareto que logró encontrar el algoritmo multiobjetivo, si durante su ejecución no encontró ningún individuo solución ($V_T = 0, V_{C_1} = 0, V_{C_2} =$

$0, \dots, V_{C_M} = 0$). Si logró encontrar la solución, en EP solo estará el individuo solución con $V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$.

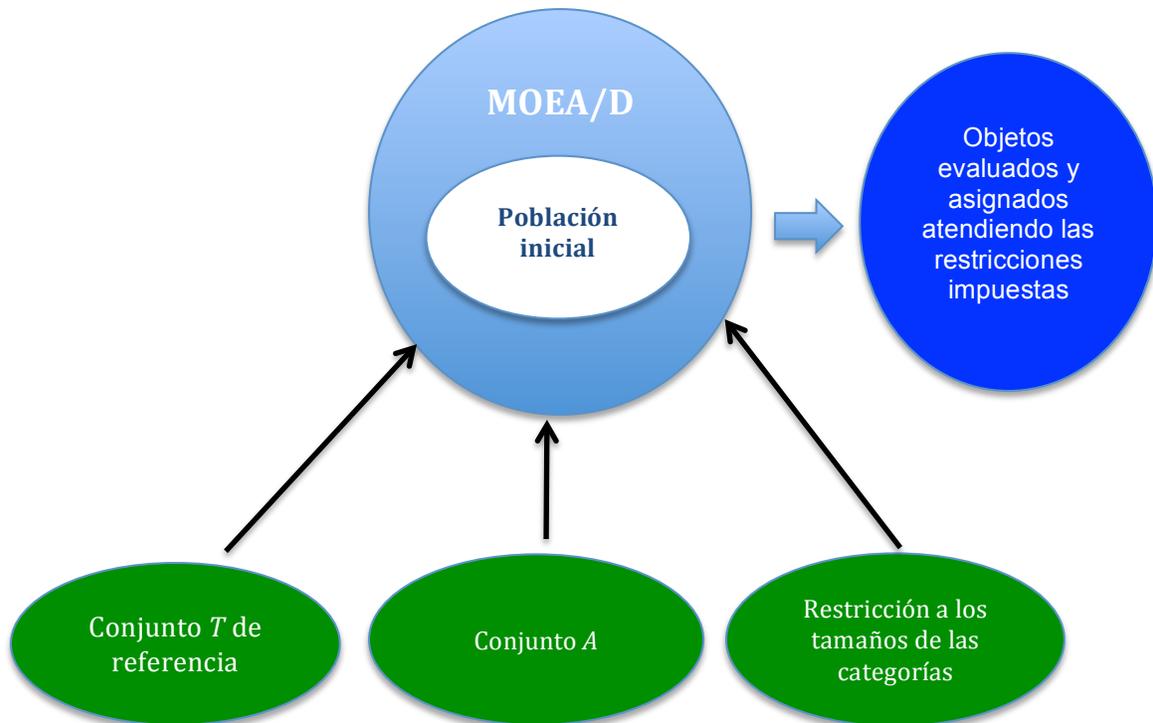


Figura 4.1. Implementación del algoritmo multiobjetivo MOEA/D sin el método PDA

4.2.1. Implementación del algoritmo MOEA/D sin utilizar el método PDA

Cuando se modela mediante optimización multiobjetivo el problema de clasificación ordinal multicriterio sin utilizar un método PDA, el algoritmo multiobjetivo MOEA/D atiende de manera simultánea las preferencias del decisor implícitas en los objetos del conjunto de referencia y las preferencias acerca del tamaño de cada una de las categorías. Utiliza como entrada los objetos del conjunto de referencia, los objetos del conjunto A y las restricciones sobre el tamaño de las categorías impuestas por el decisor. De los objetos asignados en el conjunto de referencia se conocen los desempeños en cada criterio y su clasificación, esta información es suministrada por el DM. Mientras que de los objetos del conjunto A solo se conocen los desempeños en cada criterio. El problema a resolver es cómo clasificar a estos objetos respetando las restricciones en el tamaño de las categorías impuestas por el decisor y las preferencias implícitas en los objetos asignados en el

conjunto de referencia T . En la figura 4.1 se muestra como es utilizada esta información por el algoritmo MOEA/D.

4.2.2. Implementación del algoritmo MOEA/D utilizando el método PDA

Cuando se emplea el método PDA con el algoritmo MOEA/D se realizan dos etapas principales: un Análisis de Desagregación de Preferencias (PDA) al conjunto T y la clasificación de los objetos del conjunto A . En la primera fase se realiza la inferencia de parámetros sobre la base de la información de referencia (conjunto T), para obtener los parámetros de un modelo ELECTRE TRI-B. En la segunda fase se utiliza al algoritmo multiobjetivo MOEA/D con el método de clasificación ordinal ELECTRE TRI-B para clasificar los objetos del conjunto A . La segunda fase emplea la información preferencial obtenida en la primera etapa y las restricciones sobre el tamaño de las categorías impuestas por el decisor.

Como se vio en la sección 3.2.1, el objetivo del método PDA es generar una población semilla con 50 individuos, que representan cada uno a un conjunto de parámetros preferenciales. Se espera que la información preferencial contenida en la población semilla sea de utilidad en la búsqueda de soluciones del algoritmo MOEA/D al brindar información de conjuntos preferenciales que pueden clasificar a la mayoría, incluso a la totalidad de los objetos del conjunto de referencia. Con esta suposición, el algoritmo MOEA/D recibe conjuntos preferenciales ajustados en el primer objetivo (violaciones al conjunto de referencia, V_T) y se tiene la expectativa de que el algoritmo MOEA/D pueda ajustar los parámetros preferenciales (violaciones al tamaño de las categorías, $V_{C_1}, V_{C_2}, \dots, V_{C_M}$) del modelo con mayor facilidad, que al tratar de resolver todos los objetivos de manera simultánea como en la sección 4.2.1.

4.2.2.1. Utilización de la población semilla

La población semilla generada por el algoritmo genético que implementa el método PDA de la sección 3.2 contiene los 50 mejores individuos que encontró el algoritmo genético durante su búsqueda. Estos 50 individuos pueden ser en el mejor de los casos, 50 individuos solución diferentes o 50 mejores individuos que pudo encontrar el algoritmo genético. Otra combinación posible para la población semilla es que contenga individuos

solución y otros que representen a los mejores individuos que se pudieron encontrar en la ejecución del algoritmo genético. Se define como mejor individuo al conjunto de parámetros preferenciales obtenido por el método PDA, que puede representar mejor las preferencias del decisor (clasifica correctamente al mayor número de objetos del conjunto de referencia). Un individuo solución es un conjunto de parámetros preferenciales que pueden clasificar a la totalidad de objetos del conjunto de referencia T de manera idéntica al decisor.

Cada individuo de la población semilla representa un modelo preferencial (ELECTRE TRI-B) conformado por (umbrales, pesos de los criterios, perfiles frontera y el umbral de credibilidad). La población semilla es obtenida considerando solo la información preferencial del conjunto de referencia (conjunto T). Por este motivo los individuos (que representan un modelo preferencial) solo están ajustados a las política preferencial del decisor implícita en el conjunto de referencia (pero no a la información preferencial del decisor considerada en los tamaños a las categorías). Los individuos solución encontrados y guardados en la población semilla clasifican de manera idéntica al decisor los objetos de referencia.

La hipótesis que subyace es que la población semilla tiene individuos que contienen buena información (buenos modelos preferenciales) que logren clasificar los objetos del conjunto de referencia como el decisor. Pero estos modelos no consideran la información preferencial de las restricciones a los tamaños de las categorías. De tal manera que al utilizar estos individuos semilla en la población inicial de MOEA/D, se debería obtener ayuda para el algoritmo al proporcionar el primer objetivo (V_T) con valores muy bajos de violaciones, incluso con 0 violaciones para el caso de individuos solución. La figura 4.2 muestra la manera en que la población semilla se inserta en la población inicial del algoritmo MOEA/D.

Al inicio del algoritmo MOEA/D se genera la población inicial (número de subproblemas; 451 para 4 objetivos, 600 para 5 objetivos y 800 para 6 objetivos) de manera aleatoria, posteriormente se reemplazan 50 individuos de la población inicial de MOEA/D por los 50 individuos de la población semilla. Como los individuos semilla solo están evaluados en (V_T) se calcula la aptitud de todos los individuos de la población inicial en los objetivos

$V_{C_1}, V_{C_2}, \dots, V_{C_M}$, para calcular las violaciones a las restricciones en el tamaño de cada una de las categorías.

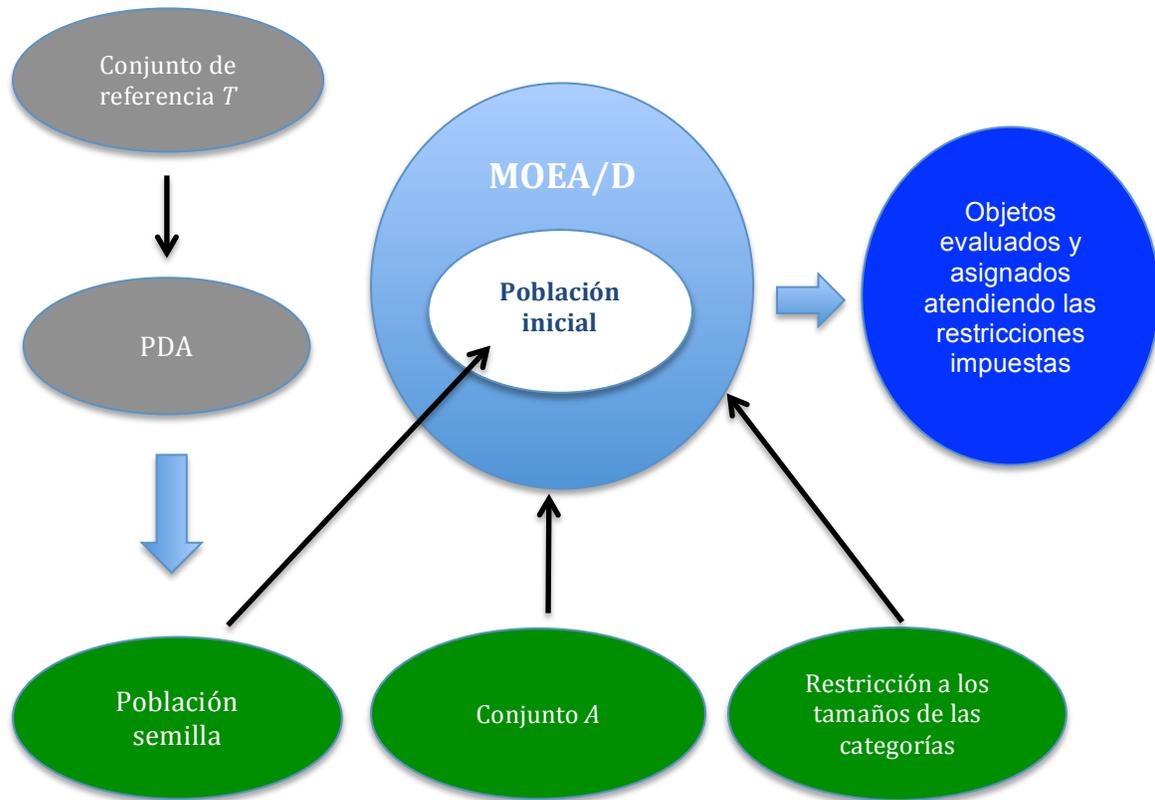


Figura 4.2. Inserción de la población semilla en la población inicial de MOEA/D

4.3. Generador de instancias

Para formar las instancias utilizadas en la experimentación se diseñó un generador al que se le proporcionan como datos de entrada una combinación de parámetros (número de categorías, número de criterios, objetos por categoría del conjunto T y número de objetos del conjunto A). Con esto, el generador proporciona una instancia como dato de salida. El generador crea un DM virtual representado por un conjunto de parámetros preferenciales, de esta manera se asegura que para la instancia creada existe al menos una solución ideal. Este conjunto de parámetros clasifica a todos los objetos del conjunto de referencia y a todos los objetos del conjunto A , respetando el tamaño de las categorías impuesto por el DM. La figura 4.3 muestra el generador diseñado.

Cada vez que el generador forma una instancia, primero crea un modelo de un decisor (umbrales, pesos de los criterios, perfiles frontera y el umbral de credibilidad), y posteriormente genera objetos utilizando este modelo hasta llenar el conjunto T y el conjunto A con los objetos requeridos. Antes de asignar un objeto al conjunto T o al conjunto A , el generador se asegura que en los conjuntos no haya ningún objeto idéntico al que se pretende ingresar.

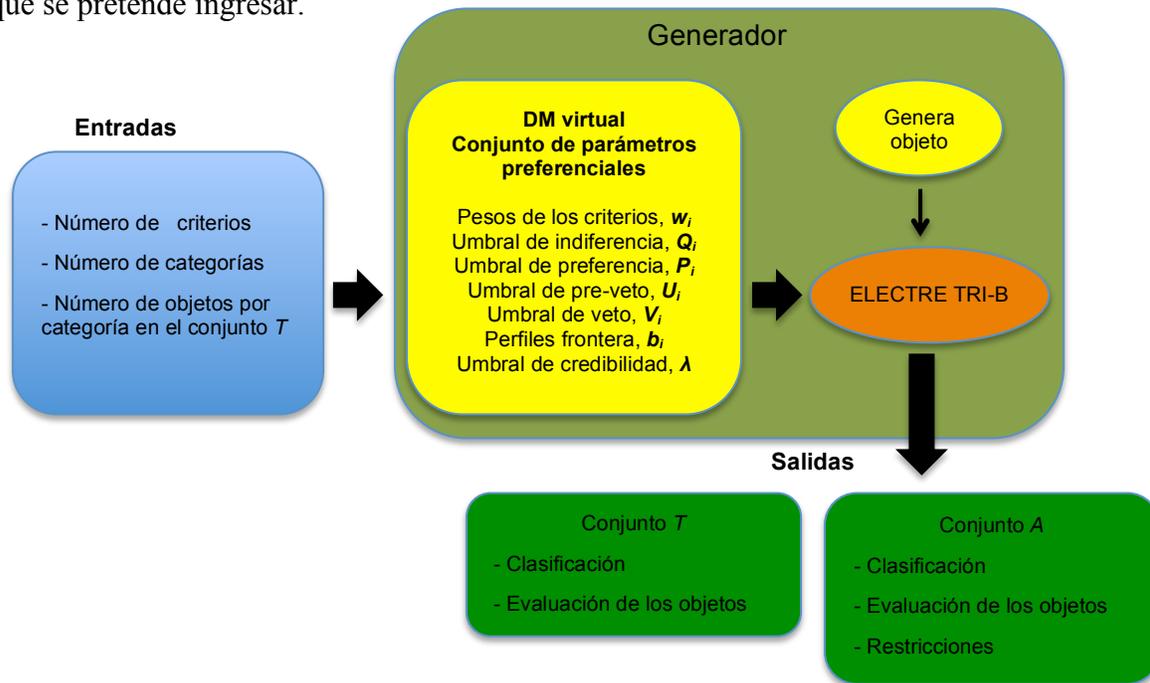


Figura 4.3. Generador de instancias para experimentación del algoritmo MOEA/D

Las restricciones en los tamaños de las categorías del conjunto A son calculadas por el generador considerando un $\pm 20\%$ del promedio de objetos para una combinación de parámetros dada. Con esto se tiene la certeza que para los datos generados existe al menos una solución ideal, que es el modelo preferencial que creó el generador.

4.4. Experimentación

Para probar la metodología propuesta se realizó una experimentación exhaustiva para diferentes tamaños de categorías, número de criterios, número de objetos en el conjunto A y número de objetos por categoría del conjunto T de referencia (vea tabla 4.1).

En un primer esquema de experimentación se probó el algoritmo MOEA/D sin utilizar el método PDA, y en una segunda configuración se utilizó el método PDA. Como se explicó en la sección 3.1 el método PDA realiza una primera etapa para inferir los parámetros preferenciales implícitos en un conjunto de referencia. El método PDA proporciona una población semilla que es utilizada en la población inicial del algoritmo MOEA/D. La hipótesis es que el método PDA debe ayudar al algoritmo MOEA/D en su búsqueda, ya que proporciona individuos que pueden clasificar a la mayoría o a la totalidad de los objetos del conjunto de referencia. Estos individuos probablemente sean mejores que los individuos generados aleatoriamente por el algoritmo MOEA/D en su población inicial.

Tabla 4.1. Valores utilizados en la experimentación

Número de categorías	Número de criterios	Objetos por categoría en conjunto T	Tamaño del conjunto A
3	3	3	50
4	5	5	100
5	7	7	200
		10	500
			1000

Con los datos de la Tabla 4.1 se forman un total de 180 combinaciones diferentes de parámetros (número de categorías, número de criterios, objetos por categoría del conjunto T y número de objetos del conjunto A). Para la experimentación se generaron 20 instancias distintas por cada una de las 180 combinaciones de parámetros posibles. Se considera una instancia a un conjunto T , un conjunto A y las restricciones a los tamaños de las categorías en que deben ser clasificados los objetos del conjunto A . En total se generaron 3600 instancias.

4.5. Resultados

En este apartado del trabajo se muestran los resultados obtenidos de la experimentación planteada en la sección 4.4. Primero se mostrarán y analizarán los resultados obtenidos por

el algoritmo MOEA/D sin utilizar el método PDA. Posteriormente expondrán los resultados obtenidos al utilizar el método PDA combinado con el algoritmo multiobjetivo MOEA/D.

Todos los experimentos fueron realizados en un equipo de cómputo con un procesador Intel core i7 de 2.6 Ghz con 8 Gb de memoria RAM, 512 GB de disco duro de estado sólido y sistema operativo macOS High Sierra.

4.5.1. Sin utilizar el método PDA

Las tablas 4.2, 4.3, 4.4, 4.5 y 4.6 muestran los resultados de los 3600 experimentos efectuados con el algoritmo MOEA/D sin utilizar el método PDA. La tabla 4.2 muestra la efectividad por número de categorías. Los valores mostrados corresponden al porcentaje de veces que el algoritmo clasificó a la totalidad de los objetos del conjunto A , cumpliendo las restricciones impuestas por el decisor. Los porcentajes de efectividad son calculados del total de los experimentos realizados por cada número de categorías. El algoritmo MOEA/D tuvo una efectividad del 100% cuando resolvió instancias con 3 categorías, y su efectividad muestra una tendencia a disminuir al aumentar el número de ellas.

Tabla 4.2. Efectividad por número de categorías sin método PDA

	3 categorías	4 categorías	5 categorías
MOEA/D	100 %	94.66 %	89.08 %

Tabla 4.3. Efectividad por número de criterios sin método PDA

	3 criterios	5 criterios	7 criterios
MOEA/D	96.58 %	94.83 %	92.33 %

La tabla 4.3 muestra un análisis del comportamiento de la efectividad por número de criterios. Para esta tabla los porcentajes de efectividad se calculan del total de los experimentos realizados por cada número de criterios utilizados para la experimentación. La efectividad del algoritmo MOEA/D se degrada al incrementar el número de criterios considerados para los objetos.

La tabla 4.4 muestra la efectividad del algoritmo MOEA/D analizando el efecto de incrementar el número de objetos a clasificar en el conjunto A . Para todos las instancias que tenían 50 objetos en el conjunto A , el algoritmo MOEA/D logró clasificar a todos los objetos atendiendo las restricciones en los tamaños de las categorías impuestas por el decisor. Lo que significa que el algoritmo fue capaz de encontrar la solución al problema (no necesariamente igual al DM virtual).

Al incrementar el número de objetos del conjunto A , la efectividad del algoritmo MOEA/D fue disminuyendo. Para esta tabla los porcentajes de efectividad se calculan del total de los experimentos realizados por cada número de objetos del conjunto A utilizados para la experimentación.

Tabla 4.4. Efectividad por tamaño del conjunto A sin método PDA

	50 objetos	100 objetos	200 objetos	500 objetos	1000 objetos
MOEA/D	99.44 %	98.89 %	97.22%	93.89 %	83.47 %

La efectividad del algoritmo MOEA/D para los diferentes valores en el número de objetos en cada categoría del conjunto de referencia se muestra en la tabla 4.5. Los valores para la efectividad mostrada en esta tabla indican el porcentaje de veces que el algoritmo MOEA/D fue capaz de clasificar a todos los objetos del conjunto T .

Tabla 4.5. Efectividad por número de objetos en cada categoría del conjunto T sin método PDA

	3 objetos	5 objetos	7 objetos	10 objetos
MOEA/D	97.33 %	96.22 %	93.89 %	90.33 %

La tabla 4.6 muestra algunos de los tiempos de ejecución que tardó el algoritmo MOEA/D para encontrar la solución. Los valores de tiempo mostrados son para 3, 4 y 5 categorías, 3 criterios, 50 y 1000 objetos del conjunto A y 3 objetos por cada categoría del conjunto T . Como se puede apreciar en esta tabla, el incremento del número de objetos del conjunto A y del número de categorías del problema de clasificación, tiene como consecuencia un aumento en el tiempo de ejecución del algoritmo MOEA/D. Los tiempos mostrados en la tabla 4.6 corresponden a la media de tiempos de ejecución de las 20 instancias diferentes

creadas por el generador para la misma configuración de parámetros (número de criterios, número de categorías, número de objetos del conjunto A y número de objetos por categoría del conjunto T).

Tabla 4.6. Tiempos de ejecución sin método PDA

Número de categorías	Número de criterios	Objetos por categoría en el conjunto T	Tamaño del conjunto A	Tiempo
3	3	3	50	10.8 s
4	3	3	50	5.38 m
5	3	3	50	8.55 m
3	3	3	1000	29.16 m
4	3	3	1000	93.86 m
5	3	3	1000	146.27 m

4.5.2. Utilizando el método PDA

En esta sección se muestran los resultados obtenidos en la experimentación al emplear la población semilla generada por el método PDA, como información de entrada para el algoritmo multiobjetivo MOEA/D.

Tabla 4.7. Efectividad por número de categorías con método PDA

	3 categorías	4 categorías	5 categorías
PDA	100 %	99.83 %	99.42 %
MOEA/D	100 %	99.08 %	98.42 %

Las tablas 4.7, 4.8, 4.9, 4.10 y 4.11 muestran los resultados de los 3600 experimentos efectuados utilizando el método PDA. La tabla 4.7 muestra la efectividad del algoritmo MOEA/D por número de categorías. Los valores mostrados para el algoritmo MOEA/D corresponden al porcentaje de veces que el algoritmo clasificó a la totalidad de los objetos del conjunto A , cumpliendo las restricciones impuestas por el decisor. Los valores para el PDA indican el porcentaje de veces cuando el procedimiento de inferencia fue capaz de

clasificar a todos los objetos del conjunto T . Los porcentajes de efectividad son calculados del total de los experimentos realizados por cada tamaño de categoría. Tanto el PDA como el algoritmo MOEA/D tuvieron una efectividad del 100% cuando resolvieron instancias con 3 categorías, y su efectividad una tendencia a disminuir al aumentar el número de categorías.

Tabla 4.8. Efectividad por número de criterios con método PDA

	3 criterios	5 criterios	7 criterios
PDA	100 %	99.58 %	99.66 %
MOEA/D	99.42 %	99.08 %	99.00 %

La tabla 4.8 muestra un análisis del comportamiento de la efectividad por número de criterios. Para esta tabla los porcentajes de efectividad se calculan del total de los experimentos realizados por cada número de criterios utilizados para la experimentación. La efectividad del método PDA y del algoritmo MOEA/D se degrada al incrementar el número de criterios considerados para los objetos. Un dato importante es que el método PDA tuvo una efectividad menor cuando se trabajó con objetos definidos por 5 criterios, que al considerar objetos con 7 criterios.

La tabla 4.9 muestra la efectividad del algoritmo MOEA/D analizando el efecto de incrementar el número de objetos a clasificar en el conjunto A . Para todas las instancias que tenían 50 objetos en el conjunto A , el algoritmo MOEA/D consiguió clasificar a todos los objetos atendiendo las restricciones en los tamaños de las categorías impuestas por el decisor. Al incrementar el número de objetos del conjunto A , la efectividad del algoritmo MOEA/D fue disminuyendo. Para esta tabla los porcentajes de efectividad se calculan del total de los experimentos realizados por cada número de objetos del conjunto A utilizados para la experimentación.

Tabla 4.9. Efectividad por tamaño del conjunto A con método PDA

	50 objetos	100 objetos	200 objetos	500 objetos	1000 objetos
MOEA/D	100 %	99.31 %	99.44 %	98.75 %	98.33 %

La efectividad del método PDA para los diferentes valores del número de objetos en cada categoría del conjunto de referencia se muestra en la tabla 4.10. Los valores para la efectividad mostrada en esta tabla indican el porcentaje de veces que el procedimiento PDA fue capaz de clasificar a todos los objetos del conjunto T .

La efectividad del método PDA implementado fue del 100% cuando se tenían 3 y 5 objetos por categoría en el conjunto de referencia T y cuando todos los objetos a clasificar eran evaluados en 3 criterios. La efectividad más baja para el PDA se presentó cuando se tenían 10 objetos por categoría del conjunto T con objetos evaluados en 7 criterios.

Tabla 4.10. Efectividad por número de objetos en cada categoría del conjunto T con método PDA

	3 objetos	5 objetos	7 objetos	10 objetos
PDA	100 %	100 %	99.67 %	99.33 %

Tabla 4.11. Tiempos de ejecución con método PDA

Número de categorías	Número de criterios	Objetos por categoría en el conjunto T	Tamaño del conjunto A	Tiempo
3	3	3	50	8.63 s
4	3	3	50	1.33 m
5	3	3	50	4.21 m
3	3	3	1000	11.56 m
4	3	3	1000	32.25 m
5	3	3	1000	42.21 m

La tabla 4.11 muestra algunos de los tiempos de ejecución que tardaron el método PDA y el algoritmo MOEA/D para encontrar la solución. Los valores de tiempo mostrados son para 3, 4 y 5 categorías; 3 criterios; 50 y 1000 objetos del conjunto A y 3 objetos por cada categoría del conjunto T . Como se puede apreciar en la tabla el incremento del número de objetos del conjunto A y del número de categorías del problema de clasificación, tiene como consecuencia un aumento en el tiempo de ejecución del método PDA y del algoritmo

MOEA/D. Los tiempos mostrados en la tabla 4.11 corresponden a la media de tiempos de ejecución de las 20 instancias diferentes creadas por el generador para la misma configuración de parámetros (número de criterios, número de categorías, número de objetos del conjunto A y número de objetos por categoría del conjunto T).

Todos los experimentos fueron realizados en un equipo de cómputo con un procesador Intel core i7 de 2.6 Ghz con 8 Gb de memoria RAM, 512 GB de disco duro de estado sólido y sistema operativo macOS High Sierra.

4.6. Comparación de resultados

Al inicio de este trabajo de investigación, se tenía la hipótesis de que utilizar un método PDA sería de ayuda para la efectividad del algoritmo de optimización multiobjetivo. El método PDA busca un modelo preferencial que clasifique los objetos del conjunto de referencia de manera idéntica al decisor. Después de la implementación del método PDA y la experimentación realizada se comprobó que la inferencia de modelos de preferencias del conjunto de referencia es de gran ayuda para el problema de optimización multiobjetivo.

En los dos casos de experimentación: sin el método PDA y con el método PDA al incrementar el número de criterios o la cardinalidad del conjunto A , la efectividad del método disminuye debido a que la complejidad del problema se incrementa.

Otro resultado importante es que el método PDA implementado mejoró la efectividad y la eficiencia del algoritmo propuesto, como se puede ver en las tablas 4.6 y 4.11. Se tenía la hipótesis de que el método PDA sería de ayuda para mejorar la efectividad del algoritmo pero se tenía la incertidumbre de que al dividir el problema en dos etapas (desagregación de preferencias y optimización multiobjetivo) pudiera tomar más tiempo de ejecución que solo utilizar el algoritmo MOEA/D para resolver el problema como, un problema de optimización multiobjetivo.

Es importante mencionar que todos los resultados de las tablas 4.2 – 4.11 consideran solamente las veces que el algoritmo MOEA/D fue capaz de encontrar la solución, es decir

un individuo con 0 violaciones en todos sus objetivos ($V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$).

4.7. Ordenamiento de los objetos al interior de las categorías

Una vez que el algoritmo multiobjetivo MOEA/D termina su ejecución proporciona una solución o un conjunto de soluciones del frente de Pareto, que se tratan de cumplir las preferencias del decisor implícitas en el conjunto de referencia y las preferencias que expresa en las restricciones al tamaño de las categorías. Los objetos en cada una de las categorías se consideran iguales, no se diferencian unos de otros. Para dar la posibilidad de diferenciar entre objetos de una misma categoría, es necesario realizar un procedimiento de ordenamiento de los objetos al interior de la categoría.

El punto central de este trabajo radica en presentar una propuesta de optimización multiobjetivo para el problema de clasificación ordinal multicriterio con restricciones al tamaño de las categorías. El procedimiento de ordenamiento puede ser realizado mediante métodos de explotación de relaciones de preferencia borrosas como el score del flujo neto de outranking (Fodor and Roubens, 1992), el método de “destilación” que emplea ELECTRE III (Ostanello, 1985), o los métodos basados en la solución de problemas de optimización multiobjetivo de inconsistencias propuestos por Fernández et al., 2008, Fernández y Leyva, 2004, y Leyva y Fernández, 1999, y más recientemente en la tesis doctoral de Gastélum Chavira (2014). La elección de método de ordenamiento a utilizar dependerá básicamente del número de objetos en cada categoría. Un método simple, que a la vez es escalable cuando aumenta el número de acciones a ordenar, es el conteo de flujo neto de outranking, utilizado por PROMETHEE (Brans y Vincke, 1985) y por Kadzinski y Slowinski (2015) en su propuesta de solución al CSP. Este método ha sido criticado porque no controla las inconsistencias entre la posición de las acciones en el ranking y las relaciones de preferencia que se derivan de la relación borrosa de no-inferioridad. Quizás la forma con mejor fundamento para el manejo de esas inconsistencias es la propuesta de Fernández et al. (2008) y de la tesis doctoral de Olmedo Pérez (2009), basada en el cumplimiento del llamado Principio de Correspondencia, y en la solución de un problema de optimización multiobjetivo mediante algoritmos evolutivos. Este método puede reducir

las violaciones de preferencias explícitas, identificándolas, controlándolas y minimizándolas. En los experimentos numéricos de Fernández et al., (2008) la propuesta mostró un buen rendimiento en el sentido de la calidad de las soluciones, y en el esfuerzo computacional que requiere. Sin embargo, solo ha sido validado experimentalmente en el ordenamiento de conjuntos de pequeña cardinalidad, y no está claro cuán bien escala el algoritmo evolutivo con el aumento del número de acciones a ordenar.

Capítulo 5.

Conclusiones

En este capítulo se presentan las conclusiones del trabajo sobre el problema de clasificación ordinal multicriterio con restricciones al número de objetos en las categorías, utilizando un enfoque evolutivo multiobjetivo. También se presenta un análisis de las respuestas a las preguntas de investigación y el trabajo futuro.

5.1. Análisis del cumplimiento de objetivos y de las respuestas a las preguntas de investigación

En el capítulo 1 de este documento de tesis se plantearon tres preguntas de investigación. Estas preguntas son consideradas auxiliares para responder la interrogante central y a partir de ellas generar las conclusiones. Sobre la base de lo encontrado durante la realización de este trabajo de investigación, a continuación se presentan cada una de ellas con su respuesta:

1. ¿Contribuye la optimización multiobjetivo a lograr mejores soluciones desde el punto de vista de las preferencias integrales del DM?

Este fue uno de los retos principales de este trabajo de investigación, ya que todos los trabajos previos en el estado del arte (vea tabla 1.2) resuelven este problema como un problema mono-objetivo (generalizan las violaciones a los tamaños de las categorías, no las manejan a nivel de categoría). En la sección 4 se muestra la implementación del problema de clasificación ordinal con restricciones en el tamaño de las categorías, como un problema multiobjetivo. Esta implementación fue realizada con éxito después de varios intentos y de utilizar diferentes algoritmos. Fue realizada una implementación con el algoritmo NSGA-II, como un problema bi-objetivo. En esta implementación se consideraban de manera global en un objetivo todas las violaciones a los tamaños de las categorías y otro para las violaciones al conjunto de referencia. Se manejaron dos objetivos debido a que NSGA-II ha probado su efectividad en dos y tres objetivos, pero

no ha mostrado buenos resultados para más de tres objetivos.

La implementación con el algoritmo NSGA-II como un problema bi-objetivo resultó tener una efectividad muy baja, además tenía el inconveniente de que manejaba las violaciones a las restricciones de los tamaños de las categorías de manera global. El manejar las violaciones a los tamaños de esta manera no permite manejar información preferencial del decisor sobre las restricciones en una categoría. Debido a la falta de resultados alentadores, esta implementación no está documentada en este trabajo.

Posteriormente se utilizó el algoritmo multiobjetivo MOEA/D, del cual se tienen buenos resultados en la literatura para problemas de muchos objetivos (más de 3 objetivos). Con el algoritmo MOEA/D se formuló el problema multiobjetivo con un objetivo por categoría y un objetivo por el conjunto de referencia. Esta formulación resulta en un problema con $M+1$ objetivos (ver sección 4). La implementación con el algoritmo MOEA/D resultó exitosa al resolver el problema multiobjetivo 4.1.

La utilización de la optimización multiobjetivo para resolver el problema de clasificación ordinal multicriterio con restricciones en el tamaño de las categorías, contribuye a obtener mejores resultados para este tipo de problemas. La formulación del problema mediante optimización multiobjetivo permite considerar de una manera más integral las preferencias del decisor.

Mediante la formulación multiobjetivo del problema se le brinda la posibilidad al DM de aprender acerca del mismo, y tener la posibilidad de reconocer posibles inconsistencias en el conjunto de entrenamiento. De tal manera que dentro de las soluciones obtenidas por el algoritmo multiobjetivo MOEA/D puede haber opciones que asignen algunos objetos de referencia de manera distinta al conjunto de entrenamiento, permitiendo con esto cumplir mejor con las restricciones a los tamaños de las categorías.

2. ¿Ayudan las heurísticas multiobjetivo a resolver el problema de clasificación ordinal multicriterio con restricciones en el tamaño de las categorías con la metodología

ELECTRE, utilizando un modelo preferencial completo (Q, P, U, V , pesos de los criterios, perfiles frontera y umbral de credibilidad) para caracterizar al DM?

El interés de este trabajo se centra en utilizar el método de clasificación multicriterio ELECTRE TRI-B con un modelo preferencial completo (pesos de los criterios, umbrales de discriminación y de veto, perfiles frontera y umbral de credibilidad) para clasificar los objetos del conjunto A . El DM proporciona sus preferencias a través de objetos clasificados en el conjunto de entrenamiento T ; estas preferencias fueron obtenidas mediante un procedimiento de inferencia de parámetros. Como se vio en la sección 2.6, inferir todos los parámetros simultáneamente en modelos ELECTRE, requiere resolver un problema no lineal con restricciones no convexas, lo que es usualmente difícil.

Con el enfoque multiobjetivo se maneja la no-linealidad y se admite la capacidad de veto y las incomparabilidades que de ella resultan. Además, con este enfoque se elimina la consideración de restricción absoluta de las preferencias, dando flexibilidad al manejar las preferencias del conjunto de entrenamiento.

El problema multiobjetivo planteado (4.1) es resuelto con éxito mediante el algoritmo multiobjetivo MOEA/D empleando un modelo preferencial completo, y utilizando el método ELECTRE TRI-B para la clasificación de los objetos. En la figura 3.3 se muestra la codificación de cada individuo utilizada en el método PDA y por el algoritmo multiobjetivo MOEA/D. Cada individuo representa un modelo preferencial completo (umbrales, pesos de los criterios, perfiles frontera y umbral de credibilidad).

Los resultados obtenidos en la experimentación exhaustiva realizada, plasmados en las tablas 4.2 - 4.11, dan muestra de la efectividad de la metodología planteada para resolver el problema de clasificación ordinal multicriterio con restricción en el tamaño de las categorías.

3. ¿Cuál es la mejor forma de resolver el problema de optimización multiobjetivo?. Resolver primero el problema de extraer las preferencias del DM implícitas en el núcleo de referencia (análisis PDA) y posteriormente incorporar las restricciones a los tamaños de las categorías; o considerar el análisis PDA y las restricciones a los tamaños de las

categorías al mismo tiempo.

Al inicio de este trabajo de investigación, se tenía la suposición de que utilizar un método PDA sería de ayuda para la efectividad del algoritmo de optimización multiobjetivo. El método PDA busca un modelo preferencial que clasifique los objetos del conjunto de referencia de manera idéntica (o lo más parecida posible) al decisor. Después de la implementación del método PDA y la experimentación realizada se comprobó que la población semilla obtenida por el procedimiento de inferencia de parámetros del modelo preferencial del conjunto de referencia es de gran ayuda para el problema de optimización multiobjetivo. Además de mejorar la efectividad del algoritmo MOEA/D, el método PDA también mejora su eficiencia como se puede ver en las tablas 4.6 y 4.11.

Al implementar el método PDA para un problema de clasificación ordinal con restricciones al número de objetos en las categorías, se deben tener en cuenta las consideraciones siguientes:

- Las inconsistencias del decisor en el conjunto de referencia. Se considera una inconsistencia cuando un objeto es clasificado por el decisor en una categoría distinta a la que un modelo general de preferencias del conjunto de referencia la ubica. Esta situación dificulta o hace imposible al algoritmo genético encontrar un modelo que clasifique de manera idéntica a la totalidad de los objetos del conjunto de referencia.
- La incapacidad del algoritmo genético para encontrar un modelo que clasifique a la totalidad de los objetos del conjunto de referencia como lo hace el decisor. Un algoritmo genético tiene varios operadores y parámetros que deben ser sintonizados y utilizados de manera adecuada y acorde al problema que se quiere resolver. Estos operadores son: operador de cruce, operador de mutación, técnica de selección y reemplazo.

Adicionalmente a esto, es muy importante elegir una buena codificación del individuo y el modo de emplear los operadores de cruce, mutación y las técnicas de selección y reemplazo sobre ellos. Una elección inadecuada de estos factores tendrá un impacto en la capacidad del algoritmo genético, y por ende en su efectividad.

- Debilidades del método de clasificación elegido. Todos los métodos de clasificación multicriterio tienen ventajas, desventajas y debilidades. La elección de método a elegir debe ser acorde a la metodología que se quiere utilizar para resolver el problema de clasificación ordinal con restricción en los tamaños de las categorías. En este trabajo de investigación se decidió utilizar el método ELECTRE TRI-B, porque tiene la capacidad de manejar información ordinal, cualitativa y con cierto grado de imprecisión, además de preferencias intransitivas y situaciones de veto e incomparabilidad.

Estas tres consideraciones son de suma importancia para lograr que el método PDA sea efectivo y de ayuda para resolver el problema propuesto.

El logro de estos resultados fue resultado de una amplia investigación, muchas pruebas, varios ajustes de los parámetros del algoritmo genético y numerosas modificaciones al modelo y a la codificación del individuo. En la sección 3 se pueden ver los detalles de esta implementación.

5.2. Conclusiones

Esta investigación aporta un enfoque novedoso a la solución de problemas de clasificación ordinal multicriterio con restricción en el tamaño de las categorías. La propuesta es el primer trabajo que plantea el problema como un problema de optimización multiobjetivo y lo resuelve con enfoques metaheurísticos.

Mediante la formulación multiobjetivo del problema el DM tiene la posibilidad de reconocer posibles inconsistencias en el conjunto de entrenamiento. De manera tal que dentro de las soluciones obtenidas por el algoritmo multiobjetivo MOEA/D puede haber opciones que asignen algunos objetos de referencia de manera distinta al conjunto de entrenamiento, permitiendo con esto, cumplir mejor con las restricciones al tamaño de las categorías.

La metodología propuesta genera como solución un modelo preferencial completo (Q, P, U, V, pesos de los criterios, perfiles frontera y umbral de credibilidad) que permite una mejor

caracterización de un DM. El modelo tiene la ventaja de representar de una manera más completa a un DM que los procedimientos implementados en la literatura para problemas CSP (vea Tabla 1.2), que, o bien utilizan modelos compensatorios con información cardinal, o eluden la consideración de efectos de veto.

De acuerdo con los resultados obtenidos, la modelación propuesta es una herramienta efectiva para resolver problemas de clasificación ordinal multicriterio con restricción en el tamaño de las categorías. Si las preferencias son compatibles con un modelo de outranking, esta propuesta es capaz de clasificar básicamente a todos los objetos del conjunto A atendiendo las restricciones de los tamaños de las categorías impuestas por el decisor.

La experimentación exhaustiva realizada en este trabajo, con diferentes valores de criterios, categorías y tamaños de los conjuntos A y T , permite probar la robustez y la efectividad de la modelación propuesta. Eso diferencia el presente trabajo de los existentes en la literatura (vea Tabla 1.2) donde se resuelven una variedad mucho menor de problemas para probar la metodología que proponen.

Otro resultado importante es que el método PDA implementado mejoró la efectividad y la eficiencia del algoritmo propuesto. Se tenía la hipótesis de que el método PDA sería de ayuda para mejorar la efectividad del algoritmo MOEA/D, pero con la incertidumbre de que al dividir el problema en dos etapas (desagregación de preferencias y optimización multiobjetivo) pudiera tomar más tiempo de ejecución que solo utilizar el algoritmo MOEA/D para resolver el problema como un problema de optimización multiobjetivo.

5.3. Trabajo futuro

Esta investigación no concluye el estudio de problemas de clasificación ordinal multicriterio con restricciones al número de objetos en las categorías. Por el contrario, brinda un nuevo horizonte de investigación para abordar este tipo de problemas con un enfoque de optimización multiobjetivo.

Queda como trabajo futuro para dar continuidad a esta investigación:

- i) Realizar pruebas con el algoritmo evolutivo MOEA/D-DE, que utiliza evolución diferencial;
- ii) La implementación de un procedimiento de interacción con el DM para incrementar la efectividad del método;
- iii) Aplicación a casos prácticos, preferiblemente tomados de la vida real, y el manejo de las inconsistencias que se deriven de su desarrollo.

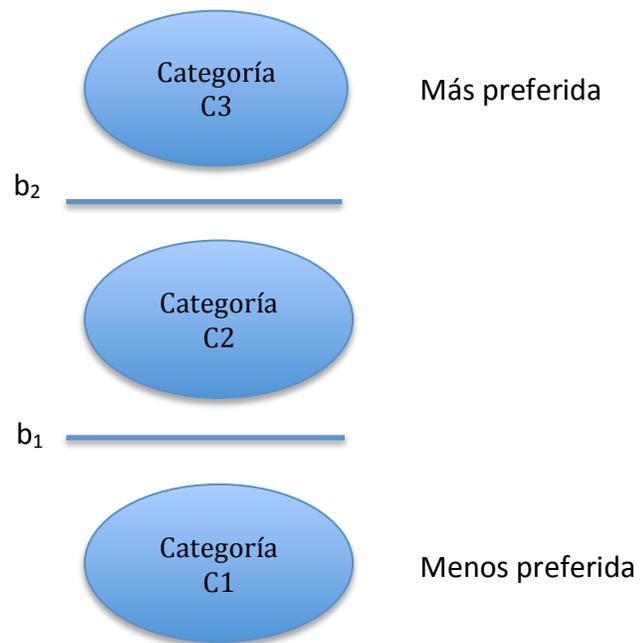
Anexo 1.

Conjunto de entrenamiento

Desempeños de los objetos

	Criterios									
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
a1	2.50	1.71	2	3.81	1.46	1.25	1.25	5	0	1
a2	4.15	3.94	2	3.93	3.58	4.00	3.75	2	0	1
a3	4.25	3.52	1	2.50	1.67	3.00	3.75	2	0	1
a4	4.00	3.72	0	3.93	2.42	3.90	3.13	2	0	1
a5	3.13	3.33	1	1.79	1.58	0.92	2.50	4	0	1
a6	2.45	1.67	0	0.71	0.83	1.00	1.25	6	0	1
a7	3.80	2.92	1	2.14	1.33	0.90	1.25	2	0	1
a8	4.25	3.50	0	2.50	2.00	2.50	3.75	2	0	1
a9	3.65	4.17	3	3.86	0.83	1.00	1.88	7	0	1
a10	4.04	3.28	0	4.21	2.08	2.20	4.29	12	0	1
a11	2.50	3.00	2	2.71	1.67	2.14	2.14	6	0	1
a12	4.50	3.89	0	2.64	1.42	3.00	3.55	12	0	1
a13	2.45	1.67	1	1.43	1.17	2.00	2.68	6	0	1
a14	3.10	2.22	1	3.57	2.17	1.20	2.50	12	0	1
a15	4.10	4.11	0	4.79	3.45	4.00	2.50	8	0	1
a16	3.72	4.21	0	4.59	0.00	0.00	3.75	6	0	1
a17	4.36	4.44	1	3.88	5.83	4.00	1.61	6	0	1
a18	3.00	4.68	4	3.37	1.67	1.00	3.75	12	0	1
a19	3.79	4.05	2	4.49	3.69	4.00	3.75	8	0	1
a20	2.22	2.94	2	2.96	0.83	2.72	2.15	10	0	1
a21	1.00	2.31	0	3.78	2.15	1.00	2.50	4	0	1
a22	3.65	4.05	0	3.99	2.38	2.00	3.75	4	0	1
a23	1.86	2.22	1	2.91	0.83	2.72	3.04	4	0	1
a24	3.00	3.33	1	3.57	2.50	2.50	3.75	8	0	1
a25	3.50	4.37	2	2.35	2.86	2.00	3.75	12	0	1
a26	3.65	3.33	2	3.78	3.46	2.00	1.25	2	0	1
a27	4.36	4.44	2	5.00	3.22	1.00	3.75	10	0	1
a28	3.93	3.33	0	5.00	0.83	1.00	3.75	2	0	1
Min	1	1.67	0	0.71	0	0	1.25	2	0	1
Max	4.5	4.68	4	5	5.83	4	4.29	12	0	1

Categorías



Clasificación

Categoría		a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28
C3	43 % (12 de 28)		■	■						■		■			■			■	■	■				■	■	■	■	■	■
C2	21 % (6 de 28)	■				■		■					■								■			■					
C1	36 % (10 de 28)				■		■		■		■		■			■	■					■	■						■

Las celdas de color verde en cada columna indica la categoría en que esta asignado el objeto. Por ejemplo, el objeto a1 esta asignado a la categoría 1 (C1) y el objeto a18 pertenece a la categoría 3 (C3).

Anexo 2.

Resultados de la experimentación sin utilizar el método PDA

Las tablas mostradas en este anexo corresponden a los resultados de la experimentación exhaustiva realizada en la sección 4.5.1. Cada tabla presenta los resultados para el número de objetos del conjunto A indicado. Las columnas de iteraciones indican la media del número de iteraciones que tardo el algoritmo multiobjetivo MOEA/D para encontrar la solución. La información plasmada en estas tablas solo considera las corridas donde el algoritmo MOEA/D encontro la solución al problema ($V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$).

Cada tabla incluye los resultados de 240 experimentos, que corresponden a las 20 repeticiones de las 12 combinaciones de los valores de número de categorías, número de criterios y número de objetos del conjunto T , que se pueden hacer por cada valor de número de objetos del conjunto A .

Las columnas de reinicios contienen la cantidad de reinicios que requirió el algoritmo MOEA/D, para encontrar la solución. La tabla de la parte inferior de cada página contiene los detalles del número de reinicios que realizó el algoritmo MOEA/D, cada vez que utilizó el procedimiento de reinicios.

3 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
3	3	3	819.88	10.8 s	0	
		5	2951.94	2.41 s	0	
		7	5691.53	0.19 s	0	
	5	3	2409.86	0.04 m	1	
		5	3227.51	1.30 m	2	
		7	6028.73	4.89 m	1	
	7	3	4287.23	1.77 m	7	
		5	4076.40	2.72 m	5	
		7	6598.34	7.17 m	3	
	10	3	5727.18	4.16 m	6	
		5	7863.23	6.38 m	8	
		7	6507.68	12.43 m	1	
						34

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
206	26	3	4	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	1	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
3	3	3	2442.47	5.13 m	1	
		5	4980.69	5.20 m	0	
		7	5119.90	6.65 m	0	
	5	3	2906.92	1.78 m	2	
		5	5056.39	4.52 m	0	
		7	8244.35	18.61 m	6	
	7	3	5719.70	3.40 m	4	
		5	10616.16	11.63 m	3	
		7	8351.60	17.01 m	10	
	10	3	7453.24	8.53 m	7	
		5	7758.97	13.91 m	5	
		7	8802.51	22.35 m	10	
						48

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
192	33	6	5	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	1	0	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
3	3	3	819.88	7.58 m	3	
		5	2951.94	13.18 m	3	
		7	5691.53	12.08 m	2	
	5	3	2409.86	9.61 m	3	
		5	3227.51	12.17 m	3	
		7	6028.73	23.83 m	3	
	7	3	4287.23	10.51 m	6	
		5	4076.40	16.34 m	6	
		7	6598.34	21.90 m	4	
	10	3	5727.18	13.33 m	9	
		5	7863.23	15.77 m	7	
		7	6507.68	31.70 m	8	
						57

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
183	42	7	4	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	2	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
3	3	3	819.88	16.56 m	3	
		5	2951.94	35.19 m	2	
		7	5691.53	34.88 m	4	
	5	3	2409.86	19.05 m	4	
		5	3227.51	9.47 m	5	
		7	6028.73	48.18 m	4	
	7	3	4287.23	35.99 m	10	
		5	4076.40	63.30 m	12	
		7	6598.34	44.97 m	13	
	10	3	5727.18	32.84 m	6	
		5	7863.23	47.02 m	11	
		7	9507.68	16.56 m	14	
						88

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
152	50	18	9	4
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	1	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
3	3	3	6146.04	29.16 m	2	
		5	7218.43	58.36 m	2	
		7	7641.31	104.37 m	2	
	5	3	6385.99	28.27 m	4	
		5	13455.47	115.84 m	11	
		7	13769.80	207.13 m	11	
	7	3	12897.36	85.89 m	13	
		5	13245.42	120.66 m	7	
		7	12456.41	178.44 m	10	
	10	3	10749.48	73.13 m	10	
		5	16366.31	165.73 m	14	
		7	16046.61	261.75 m	12	
						98

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
142	48	22	13	3
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
3	2	0	3	2
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

4 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
4	3	3	7163.30	5.38 m	6	
		5	6625.05	8.04 m	3	
		7	10422.91	16.24 m	1	
	5	3	5516.02	5.15 m	12	
		5	6087.55	6.50 m	5	
		7	9778.20	19.63 m	2	
	7	3	5972.71	7.60 m	9	
		5	11904.80	17.01 m	2	
		7	9502.68	20.53 m	1	
	10	3	15246.10	22.62 m	15	
		5	13343.07	23.00 m	3	
		7	14938.64	39.84 m	3	
						62

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
178	31	16	5	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	4	2	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	1

4 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
4	3	3	6681.99	8.73 m	2	
		5	7685.70	16.18 m	5	
		7	7063.80	20.25 m	9	
	5	3	11034.70	16.91 m	5	
		5	9685.61	20.15 m	9	
		7	8693.70	27.68 m	12	
	7	3	9419.39	14.88 m	9	
		5	9783.00	23.32 m	7	
		7	9817.31	26.06 m	10	
	10	3	11064.37	23.65 m	6	
		5	12690.13	27.90 m	9	
		7	14604.47	36.05 m	12	
						95

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
139	51	23	12	5
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	3	0	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	2	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	1	0	1

4 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
4	3	3	8848.02	27.62 m	9	
		5	9339.03	31.67 m	10	
		7	9704.42	51.19 m	11	
	5	3	9837.39	23.70 m	7	
		5	10899.13	39.77 m	10	
		7	12979.28	53.22 m	11	
	7	3	11531.73	32.57 m	8	
		5	10573.39	46.19 m	10	
		7	14987.35	69.68 m	14	
	10	3	14057.06	46.60 m	9	
		5	34593.83	153.43 m	14	
		7	46376.30	275.51 m	15	
						128

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
105	62	37	17	9
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	2	0	2	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	1	0	1	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	0	0	1	0

4 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
4	3	3	29410.97	57.78 m	17	
		5	22360.01	163.45 m	16	
		7	28967.71	352.14 m	18	
	5	3	18738.23	86.98 m	14	
		5	24587.04	183.29 m	16	
		7	19706.01	239.29 m	17	
	7	3	21059.75	114.39 m	13	
		5	32494.42	179.25 m	15	
		7	57556.02	306.93 m	19	
	10	3	24342.41	190.25 m	11	
		5	35844.38	380.56 m	15	
		7	62633.73	484.12 m	20	
						191

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
35	87	46	22	11
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
7	6	3	2	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	2	3	2	2
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
2	0	1	0	7

4 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
4	3	3	45440.95	93.86 m	19	
		5	82723.07	133.38 m	20	
		7	50895.58	696.52 m	20	
	5	3	44049.27	243.79 m	20	
		5	57026.10	414.44 m	19	
		7	41186.75	886.50 m	20	
	7	3	51507.14	458.71 m	19	
		5	125149.98	962.17 m	20	
		7	253378.32	1598.83 m	20	
	10	3	65273.48	876.66 m	20	
		5	112954.77	1344.61 m	20	
		7	231570.03	2296.89 m	20	
						237

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
3	118	53	24	11
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
5	4	4	1	2
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	3	2	2	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
4	1	1	0	1

5 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
5	3	3	9134.75	8.55 m	5	
		5	13621.86	18.70 m	8	
		7	17618.39	27.70 m	9	
	5	3	7263.76	16.69 m	4	
		5	14893.15	29.81 m	5	
		7	38264.78	38.03 m	9	
	7	3	17514.59	45.32 m	7	
		5	31653.64	67.50 m	9	
		7	58792.74	123.94 m	12	
	10	3	31593.66	137.41 m	8	
		5	53770.39	254.28 m	13	
		7	90720.13	301.60 m	18	
						107

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
133	46	27	11	5
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	3	1	3	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
2	0	3	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
2	1	0	1	0

5 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
5	3	3	6903.84	18.82 m	9	
		5	9643.72	40.85 m	10	
		7	10746.61	43.11 m	12	
	5	3	9157.19	27.59 m	10	
		5	15821.97	64.30 m	11	
		7	17160.10	88.29 m	12	
	7	3	6127.20	40.51 m	9	
		5	13556.03	71.82 m	12	
		7	15333.07	202.37 m	10	
	10	3	11989.58	148.98 m	11	
		5	23492.57	269.19 m	12	
		7	39243.34	402.25 m	13	
						131

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
109	62	31	13	6
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
4	5	2	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	1	0	2	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	0	2	1

5 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
5	3	3	10203.01	40.20 m	9	
		5	8892.08	56.26 m	7	
		7	13224.46	107.69 m	13	
	5	3	22991.83	111.04 m	12	
		5	28359.10	100.72 m	14	
		7	15854.01	243.73 m	16	
	7	3	13310.44	99.54 m	14	
		5	14683.94	200.81 m	12	
		7	21734.93	307.32 m	17	
	10	3	11745.82	165.16 m	13	
		5	36247.57	325.00 m	16	
		7	87829.05	1274.74 m	18	
						161

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
77	79	43	17	8
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
3	2	1	2	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	1	0	1
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
2	0	1	0	2

5 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
5	3	3	9916.70	81.65 m	17	
		5	14042.00	130.34 m	18	
		7	49427.35	240.06 m	20	
	5	3	16264.00	108.21 m	14	
		5	60605.42	177.54 m	19	
		7	90371.05	314.44 m	20	
	7	3	29129.67	126.67 m	13	
		5	64224.54	228.10 m	15	
		7	32083.15	832.84 m	20	
	10	3	16878.46	384.54 m	11	
		5	150255.27	2365.23 m	16	
		7	246247.12	3665.75 m	20	
						203

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
37	107	43	21	9
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
3	3	1	3	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
2	2	1	0	1
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	1	0	0	4

5 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría del conjunto T	Criterios	Iteraciones		Reinicios	
			MOEA/D	Tiempo	MOEA/D	
5	3	3	13010.80	146.27 m	20	
		5	28126.00	249.04 m	20	
		7	58037.23	773.61 m	20	
	5	3	23507.26	153.13 m	20	
		5	46195.15	588.15 m	20	
		7	69317.22	1040.59 m	20	
	7	3	128306.24	789.02 m	20	
		5	251647.00	1766.22 m	20	
		7	348009.32	4112.61 m	20	
	10	3	201205.69	2106.02 m	20	
		5	271568.77	3260.09 m	20	
		7	424146.08	5330.24 m	20	
						240

Reinicios

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
0	110	34	22	15
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
10	6	5	3	5
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	4	1	4	2
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	2	3	3	9

Anexo 3.

Resultados de la experimentación utilizando el método PDA

Las tablas mostradas en este anexo corresponden a los resultados de la experimentación exhaustiva realizada en la sección 4.5.2. Cada tabla presenta los resultados para el número de objetos del conjunto A indicado. Las columnas de iteraciones indican la media del número de iteraciones que tardo el algoritmo genético que implementa el método PDA y, el número de iteraciones que tardo el algoritmo multiobjetivo MOEA/D para encontrar la solución. La información plasmada en estas tablas solo considera las corridas de los algoritmos cuando el algoritmo MOEA/D encontro la solución al problema ($V_T = 0, V_{C_1} = 0, V_{C_2} = 0, \dots, V_{C_M} = 0$).

Cada tabla incluye los resultados de 240 experimentos, que corresponden a las 20 repeticiones de las 12 combinaciones de los valores de número de categorías, número de criterios y número de objetos del conjunto T , que se pueden hacer por cada valor de número de objetos del conjunto A .

Las columnas de reinicios contienen la cantidad de reinicios que requirieron el algoritmo genético que implementa el método PDA y el algoritmo MOEA/D, para encontrar la solución. Las dos tablas de la parte inferior de cada página contienen los detalles del número de reinicios que realizaron los algoritmos, cada vez que utilizaron el procedimiento de reinicios.

3 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
3	3	3	7844.19	13.62	8.63 s	0	0	
		5	7727.70	30.04	15.13 s	1	0	
		7	11760.05	99.43	74.92 s	0	0	
	5	3	16669.96	35.62	1.57 m	0	1	
		5	25366.95	70.08	1.94 m	0	2	
		7	37667.91	138.38	2.31 m	1	0	
	7	3	19653.98	160.00	1.84 m	3	1	
		5	45263.13	196.16	3.73 m	5	1	
		7	40451.11	278.14	4.39 m	2	2	
	10	3	28717.73	235.86	1.71 m	3	2	
		5	40264.47	853.53	2.64 m	5	0	
		7	61387.56	1326.04	3.72 m	5	0	
							25	9

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
215	10	7	3	2
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	2	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
231	8	0	1	0
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
3	3	3	8542.46	56.16	1.01 m	0	0
		5	10325.05	72.05	2.58 m	0	1
		7	9131.91	177.76	5.03 m	1	2
	5	3	13124.86	65.24	2.02 m	1	1
		5	17075.58	91.84	3.77 m	0	1
		7	32366.71	118.99	6.93 m	3	1
	7	3	13301.05	85.41	0.64 m	1	1
		5	34289.39	184.22	2.38 m	3	2
		7	42143.94	217.83	10.30 m	4	2
	10	3	37966.67	240.52	3.04 m	4	4
		5	55471.52	360.17	8.43 m	3	2
		7	138837.62	1320.70	9.24 m	6	0
						26	17

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
214	15	6	1	3
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	1	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
223	12	3	0	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	0	0	0	0

3 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
3	3	3	9198.02	416.74	1.68 m	0	0	
		5	8946.40	861.37	1.53 m	0	1	
		7	13246.77	1313.47	8.67 m	1	2	
	5	3	13142.26	455.62	5.08 m	1	2	
		5	17336.10	971.59	6.42 m	3	4	
		7	15417.19	1109.09	4.99 m	3	2	
	7	3	19896.22	744.17	5.65 m	1	4	
		5	28770.06	951.56	11.76 m	3	1	
		7	37943.78	1342.27	11.12 m	5	2	
	10	3	27557.73	917.59	7.61 m	3	5	
		5	66986.74	1452.02	9.08 m	4	6	
		7	206987.51	3766.90	16.69 m	9	7	
							33	36

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
207	16	11	1	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	2	0	2
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
204	24	8	0	4
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones		Tiempo	Reinicios		
			PDA	MOEA/D		PDA	MOEA/D	
3	3	3	8267.29	657.59	3.84 m	0	1	
		5	9483.51	955.18	14.53 m	0	4	
		7	12395.14	1059.98	11.45 m	0	2	
	5	3	13228.20	942.87	6.22 m	0	2	
		5	9872.56	1070.94	8.10 m	3	3	
		7	25025.93	2129.66	19.44 m	6	6	
	7	3	18002.21	1435.00	13.30 m	1	4	
		5	41326.21	1623.87	12.43 m	2	4	
		7	129354.84	2280.17	22.95 m	4	7	
	10	3	50991.24	1513.48	11.88 m	1	5	
		5	108855.52	3464.89	15.22 m	5	6	
		7	138381.01	5090.14	26.59 m	7	11	
							29	55

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
211	17	8	1	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
185	32	15	2	2
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	1	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

3 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
3	3	3	9413.62	467.85	11.56 m	0	5	
		5	11881.98	388.43	12.67 m	0	3	
		7	9253.36	641.41	69.41 m	0	2	
	5	3	19168.47	587.35	19.71 m	5	1	
		5	20148.94	924.33	33.06 m	4	8	
		7	33014.15	1317.66	35.58 m	4	7	
	7	3	13654.49	600.37	21.96 m	1	3	
		5	37726.43	1442.24	33.26 m	3	4	
		7	75087.83	1984.08	38.88 m	6	7	
	10	3	17417.04	219.21	14.26 m	1	3	
		5	87825.09	951.78	29.21 m	5	8	
		7	81830.14	1204.32	43.73 m	4	10	
							33	61

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
207	24	5	0	2
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	1	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
179	32	16	3	3
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	2	0	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

4 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
4	3	3	13752.84	44.14	1.33 m	1	0
		5	18754.70	52.60	2.77 m	0	2
		7	94802.77	28.19	7.42 m	3	3
	5	3	29255.33	97.02	3.57 m	2	6
		5	98176.14	143.56	5.62 m	2	0
		7	127058.75	217.33	10.37 m	8	3
	7	3	33823.95	117.62	4.38 m	5	3
		5	66264.25	190.31	14.25 m	6	0
		7	146853.89	239.01	17.47 m	9	3
	10	3	80849.25	196.25	8.33 m	7	8
		5	439925.48	362.74	18.77 m	11	5
		7	542724.33	910.10	24.31 m	3	1
						57	34

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
183	22	15	10	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	2	0	3	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	1	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	1	1	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
206	27	4	1	0
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	1	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

4 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
4	3	3	15954.18	770.81	2.61 m	0	1
		5	25117.93	935.98	6.33 m	2	3
		7	43089.74	1330.19	8.81 m	2	3
	5	3	44494.02	1104.69	3.55 m	5	4
		5	110315.46	1456.86	7.06 m	9	4
		7	60489.83	1916.00	11.27 m	9	3
	7	3	111949.31	1213.61	6.28 m	6	2
		5	107826.24	1882.48	21.31 m	9	3
		7	108111.21	2162.82	32.58 m	9	6
	10	3	117284.23	2017.93	16.62 m	10	1
		5	189154.76	4860.40	23.95 m	12	5
		7	533832.17	2770.54	33.79 m	1	0
						74	35

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
166	35	15	11	7
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	3	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	2

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
205	32	2	0	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
0	0	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	0

4 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
4	3	3	35813.60	119.28	3.90 m	3	5
		5	19835.78	561.39	25.26 m	0	9
		7	38127.56	1199.25	26.12 m	4	8
	5	3	32307.37	507.06	8.16 m	1	8
		5	43090.04	949.78	11.33 m	6	4
		7	55715.66	1071.39	27.12 m	8	5
	7	3	57117.85	1210.11	17.11 m	6	7
		5	101677.74	1580.40	31.16 m	7	9
		7	197534.70	4276.45	56.53 m	11	10
	10	3	92672.65	2336.10	12.24 m	3	3
		5	106047.31	4552.31	25.83 m	8	3
		7	606570.59	9041.76	110.89 m	4	2
						61	73

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
179	27	14	2	5
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	3	1	3	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	1	1
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	1

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
167	48	17	5	0
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	0	0	1	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	2

4 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones		Tiempo	Reinicios	
			PDA	MOEA/D		PDA	MOEA/D
4	3	3	20205.26	430.92	11.47 m	3	8
		5	38596.99	979.48	16.46 m	2	12
		7	45898.76	6089.85	72.87 m	3	10
	5	3	43149.69	713.61	24.86 m	6	14
		5	73995.53	1787.86	26.48 m	5	11
		7	141878.13	3060.44	39.63 m	7	13
	7	3	119967.44	3816.93	52.77 m	9	13
		5	283797.68	4840.78	68.93 m	7	14
		7	654359.27	9312.05	94.98 m	10	15
	10	3	436202.22	9748.98	29.23 m	9	12
		5	1612397.07	13062.46	177.65 m	9	15
		7	2462568.30	37397.79	332.76 m	12	17
						82	154

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
179	29	9	5	7
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
4	2	1	1	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	2

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
116	55	21	14	12
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
4	5	3	1	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	2	1	1	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	0	0	1	2

4 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría	Criterios	Iteraciones		Tiempo	Reinicios	
			PDA	MOEA/D		PDA	MOEA/D
4	3	3	16391.24	2898.25	32.25 m	7	14
		5	20691.19	18023.64	210.32 m	5	17
		7	28227.38	34841.18	519.33 m	9	16
	5	3	35086.43	5391.25	122.14 m	7	10
		5	76579.54	7521.67	79.56 m	9	11
		7	101831.57	13318.05	206.63 m	12	14
	7	3	432645.89	5708.30	109.43 m	7	12
		5	948028.47	17625.07	351.40 m	8	12
		7	1213027.18	49780.65	1121.46 m	11	16
	10	3	822982.36	61034.23	400.52 m	10	15
		5	2130257.78	96442.91	934.29 m	9	18
		7	3732609.38	134474.16	2485.13 m	14	22
						108	177

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
130	55	24	14	8
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	1	0	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	1	1	0	1

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
52	66	47	32	5
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
7	5	3	5	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
3	0	2	0	2
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	0	1	8

5 categorías y 50 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
5	3	3	27515.41	584.39	4.21 m	3	1	
		5	30629.32	1489.28	6.44 m	2	3	
		7	40037.46	5893.71	19.51 m	7	6	
	5	3	35184.21	302.34	3.16 m	5	1	
		5	83086.37	667.61	28.86 m	8	3	
		7	51009.61	3241.26	35.85 m	7	6	
	7	3	83043.56	834.87	36.89 m	2	3	
		5	1155400.21	1415.88	45.07 m	4	5	
		7	282948.04	4524.96	92.81 m	9	7	
	10	3	193017.21	1114.95	75.67 m	4	3	
		5	772960.29	3278.22	131.93 m	8	6	
		7	1064393.48	6122.62	251.21 m	12	8	
							71	52

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
162	31	20	12	4
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	0	1	1	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	0	0	0

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
188	27	12	5	2
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	0	0	0	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
1	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	0	1	0

5 categorías y 100 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
5	3	3	20484.17	734.31	7.55 m	4	1	
		5	190710.33	1067.70	59.62 m	5	3	
		7	370357.58	2326.15	74.29 m	9	7	
	5	3	39460.65	823.19	12.03 m	6	1	
		5	313277.29	1459.50	11.72 m	7	3	
		7	739722.41	2630.65	29.54 m	9	7	
	7	3	54375.86	1048.60	22.34 m	4	4	
		5	465117.99	2497.99	52.57 m	7	6	
		7	811286.56	12956.89	100.25 m	10	8	
	10	3	77027.51	2414.96	103.23 m	6	3	
		5	1367735.50	5116.69	196.92 m	8	8	
		7	3586926.69	17282.34	349.28 m	11	9	
							86	60

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
151	37	25	11	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	4
2	1	2	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	1	1	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	1	0	1	1

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
180	27	19	8	1
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	0	1	0	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	1	0	1

5 categorías y 200 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios		
			PDA	MOEA/D	Tiempo	PDA	MOEA/D	
5	3	3	34958.85	821.47	21.26 m	6	5	
		5	98355.57	1621.28	30.70 m	7	8	
		7	231000.72	2513.93	75.23 m	9	9	
	5	3	46209.69	1484.23	28.23 m	8	5	
		5	166217.24	2739.22	49.02 m	7	5	
		7	486352.79	4901.21	191.71 m	9	10	
	7	3	82014.26	2933.71	69.70 m	6	5	
		5	313483.26	4971.97	193.15 m	9	7	
		7	908172.68	7035.07	274.23 m	11	9	
	10	3	187307.29	4629.91	94.59 m	7	9	
		5	2563610.07	12654.23	278.82 m	9	10	
		7	6845017.52	24851.42	1111.23 m	13	14	
							101	96

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
135	52	27	12	6
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	2	0	1	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	1	0	1

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
141	61	18	11	4
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	0	0	1	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	1
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	1	1

5 categorías y 500 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
5	3	3	46836.86	886.56	33.77 m	7	11
		5	67584.00	1927.00	62.54 m	5	13
		7	126874.95	2325.93	150.69 m	10	16
	5	3	418964.00	2115.00	50.23 m	9	11
		5	532574.00	4891.00	89.34 m	11	14
		7	977745.00	8441.00	253.13 m	12	18
	7	3	843142.00	4967.00	76.49 m	9	13
		5	1243205.90	9688.99	131.95 m	12	14
		7	1792590.05	18878.87	697.95 m	13	17
	10	3	1290088.17	14429.73	210.88 m	11	14
		5	2644495.34	78552.67	2067.81 m	12	17
		7	3736251.51	111659.15	2825.77 m	15	23
						126	181

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
110	63	35	13	5
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
2	3	1	1	1
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	1	1	1
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
1	0	1	0	1

MOEA/D				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
57	86	37	26	10
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
3	4	2	3	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
2	2	0	1	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	1	3	2

5 categorías y 1000 objetos en el conjunto A

Categorías	Objetos por categoría en el conjunto T	Criterios	Iteraciones			Reinicios	
			PDA	MOEA/D	Tiempo	PDA	MOEA/D
5	3	3	27484.00	2565.00	42.21 m	9	16
		5	54393.14	3231.40	134.23 m	7	19
		7	54309.89	5732.91	564.98 m	11	18
	5	3	920542.79	3230.37	65.16 m	9	12
		5	559254.00	6429.00	486.21 m	12	13
		7	559254.00	9429.00	852.49 m	14	16
	7	3	559254.00	8329.00	678.23 m	10	14
		5	314752.27	27854.34	1143.82 m	12	15
		7	8675432.00	51647.00	3609.21 m	15	21
	10	3	559254.00	16429.00	1119.21 m	14	17
		5	925893.13	99717.28	2729.49 m	14	19
		7	1736397.26	218016.89	4242.54 m	18	25
						145	205

Reinicios

PDA				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
93	9	0	1	0
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	2	0	2	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
0	0	0	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	0	0	0	2

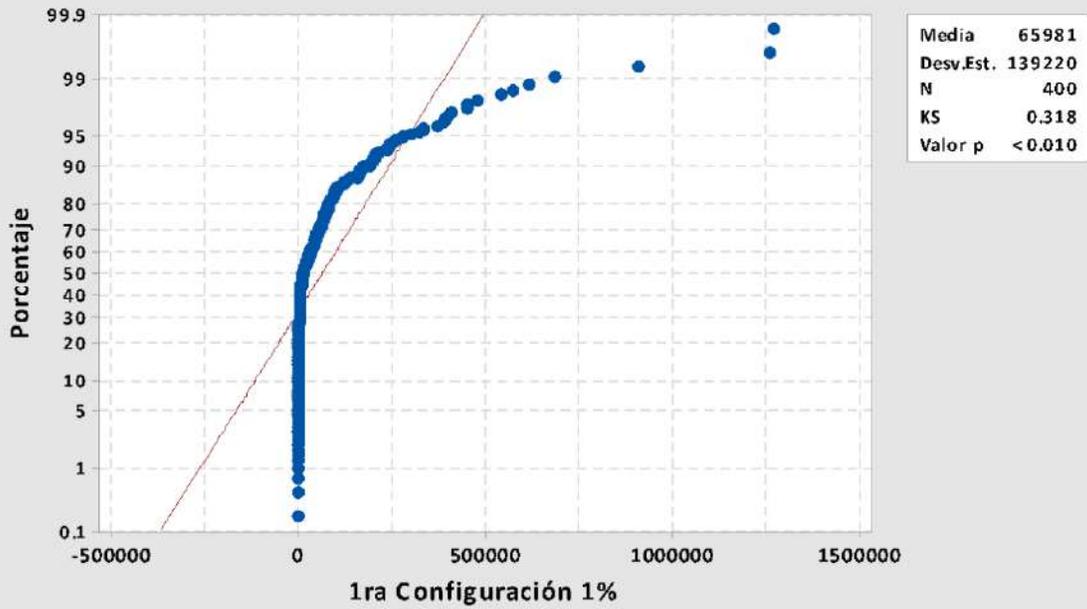
MOEAD				
Sin reinicio	1 Reinicio	2 Reinicios	3 Reinicios	4 Reinicios
28	123	51	25	2
5 Reinicios	6 Reinicios	7 Reinicios	8 Reinicios	9 Reinicios
1	2	0	1	0
10 Reinicios	11 Reinicios	12 Reinicios	13 Reinicios	14 Reinicios
2	0	1	0	0
15 Reinicios	16 Reinicios	17 Reinicios	18 Reinicios	19 Reinicios
0	1	0	2	1

Anexo 4.

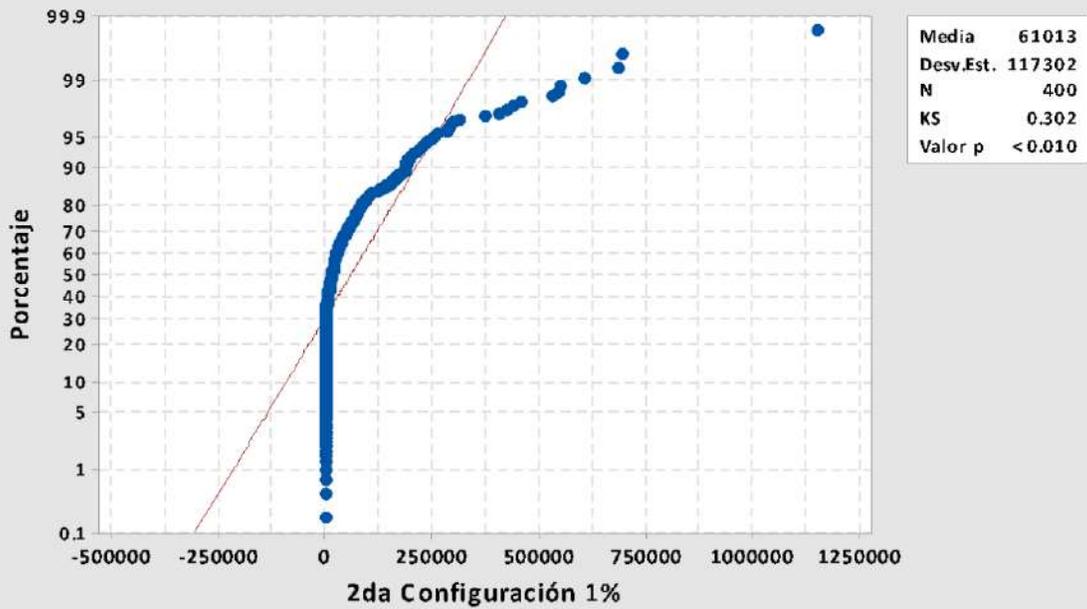
Prueba no paramétrica de Mann-Whitney a las tres configuraciones de selección y reemplazo del método PDA.

Este anexo contiene el detalle de las pruebas realizadas a los resultados de las tres configuraciones de selección y reemplazo probadas (vea tabla 3.3) en la experimentación. Las gráficas de las páginas siguientes son la prueba de normalidad realizada a cada conjunto de resultados, para revisar que los datos no presentan una distribución normal. Posteriormente, se muestran los resultados obtenidos al realizar la prueba no paramétrica de Mann-Whitney con el software Minitab.

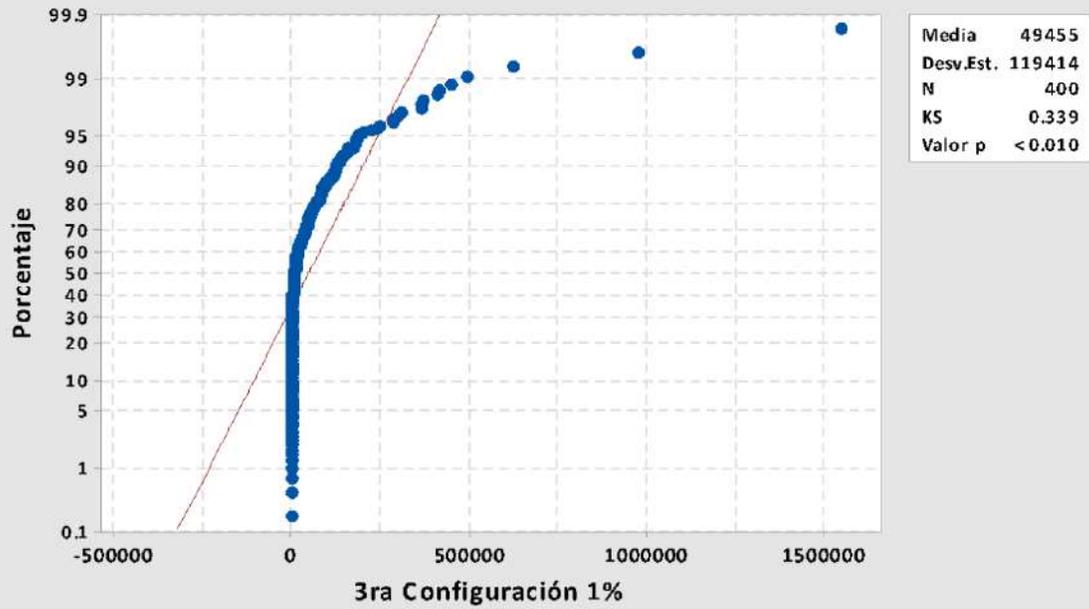
Gráfica de probabilidad de 1ra Configuración 1%
Normal



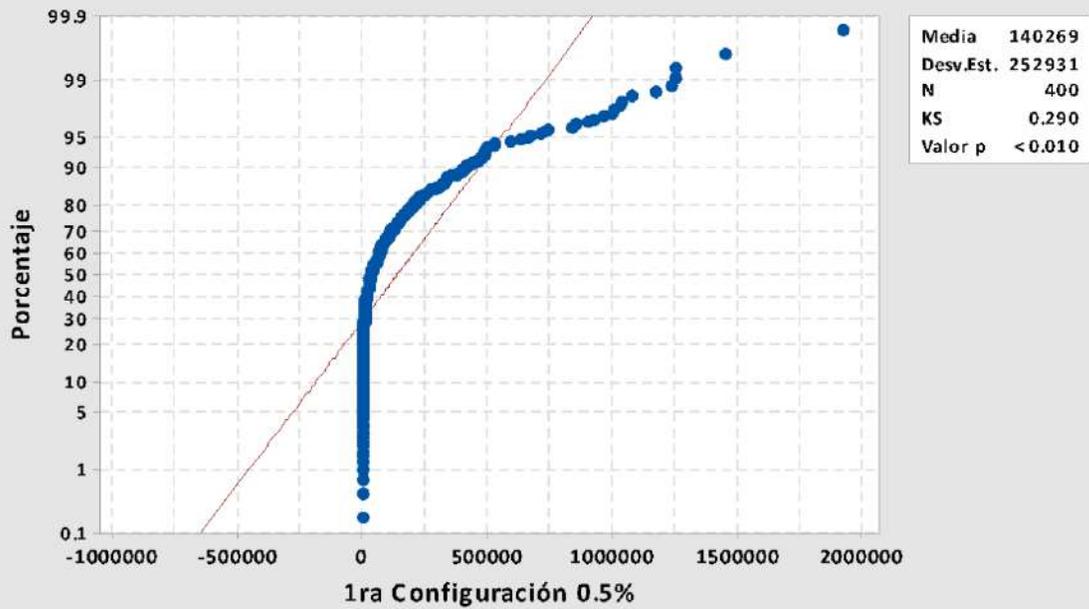
Gráfica de probabilidad de 2da Configuración 1%
Normal



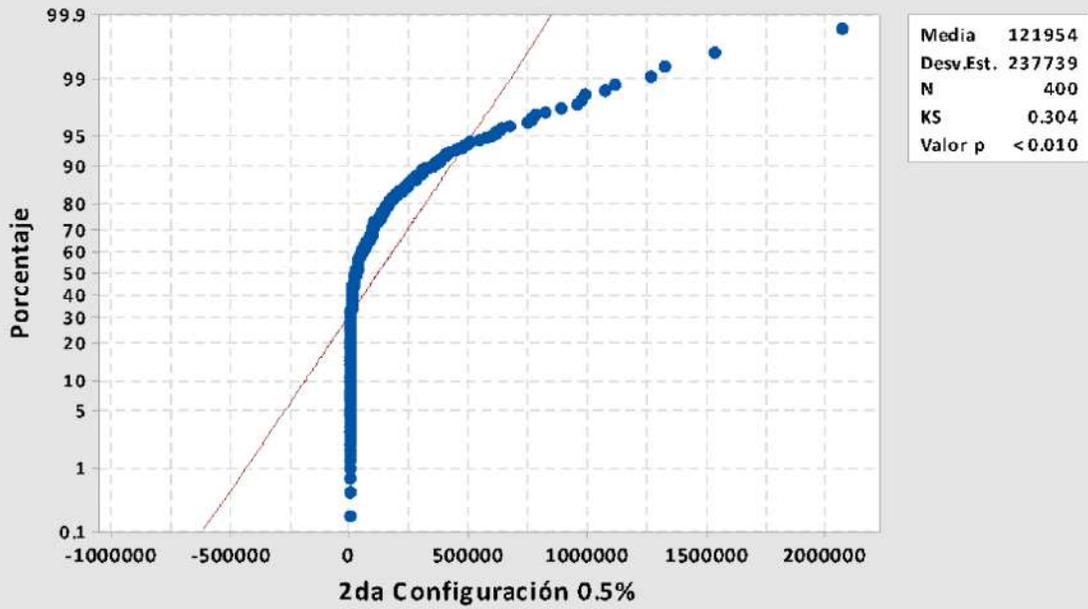
Gráfica de probabilidad de 3ra Configuración 1%
Normal



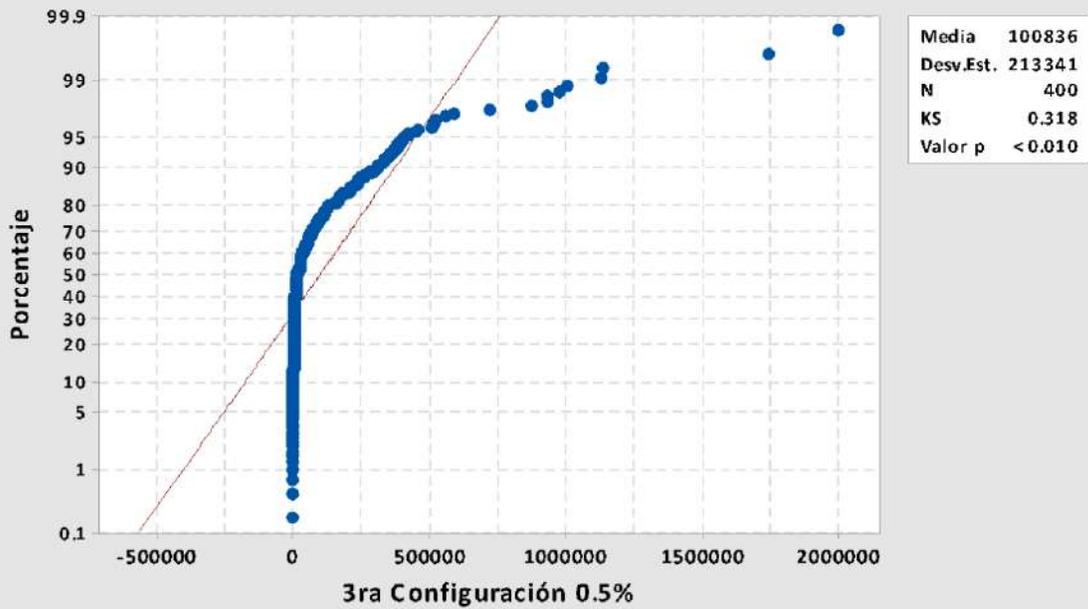
Gráfica de probabilidad de 1ra Configuración 0.5%
Normal



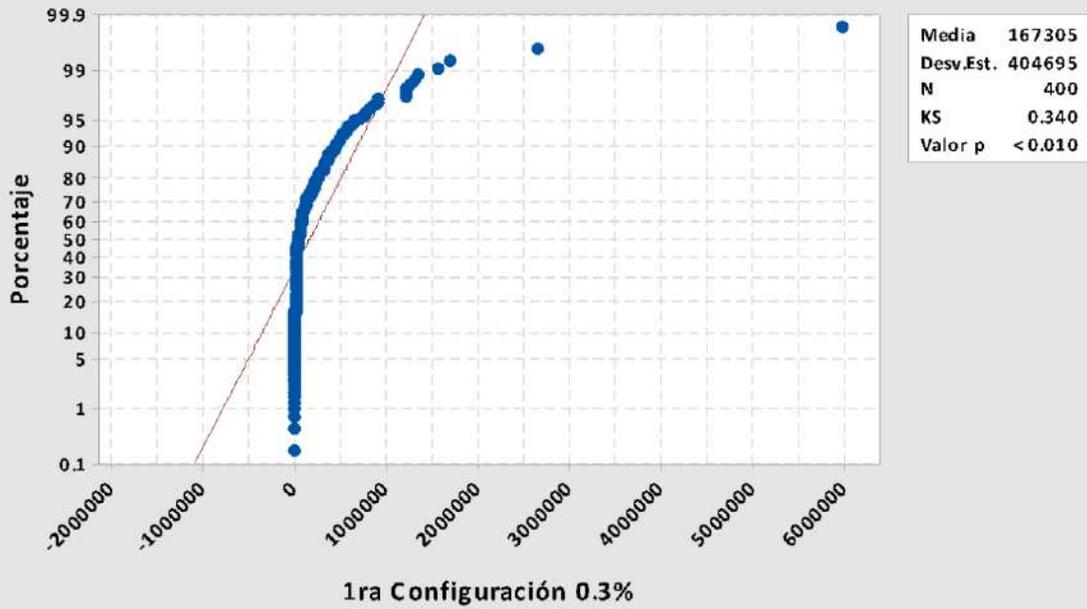
Gráfica de probabilidad de 2da Configuración 0.5%
Normal



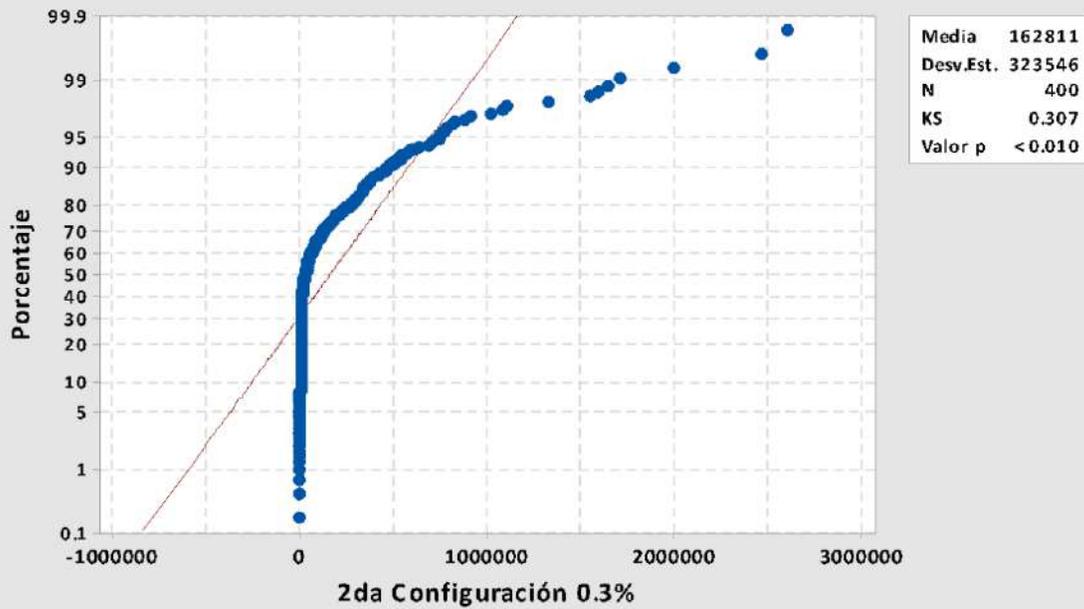
Gráfica de probabilidad de 3ra Configuración 0.5%
Normal



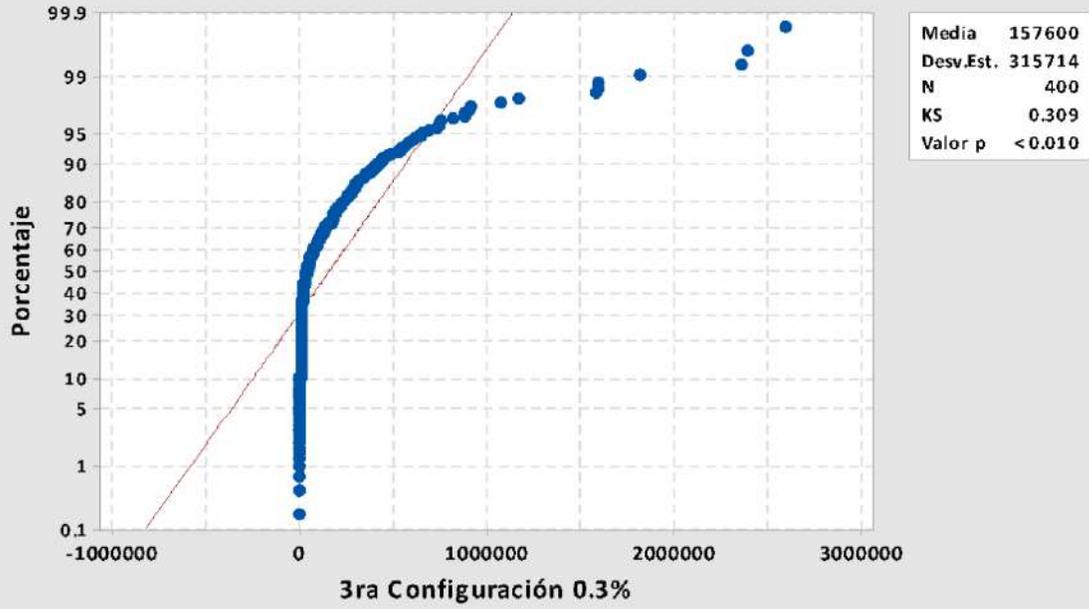
Gráfica de probabilidad de 1ra Configuración 0.3%
Normal



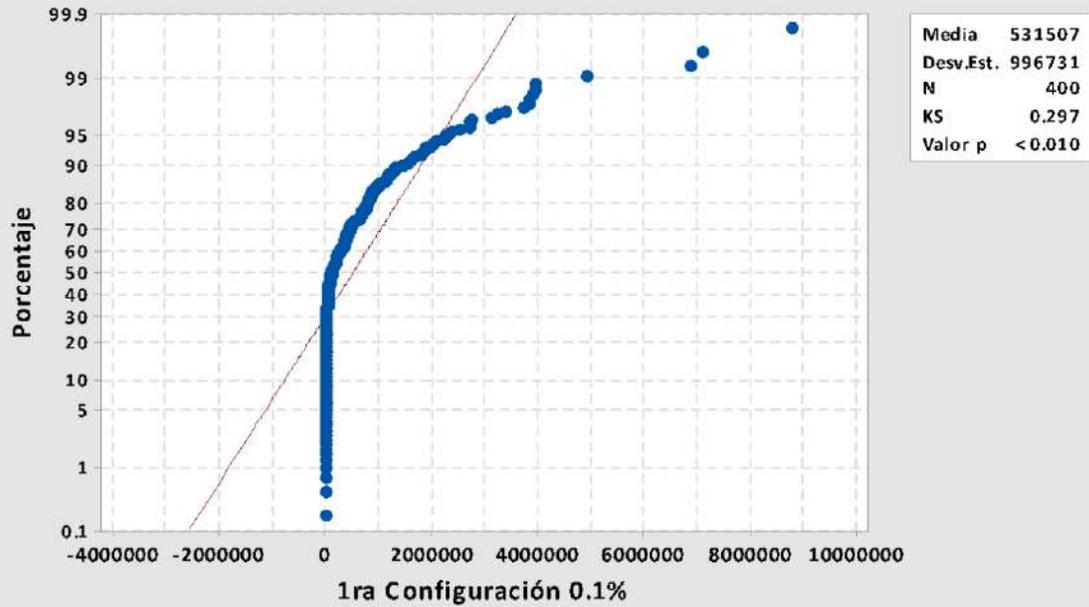
Gráfica de probabilidad de 2da Configuración 0.3%
Normal



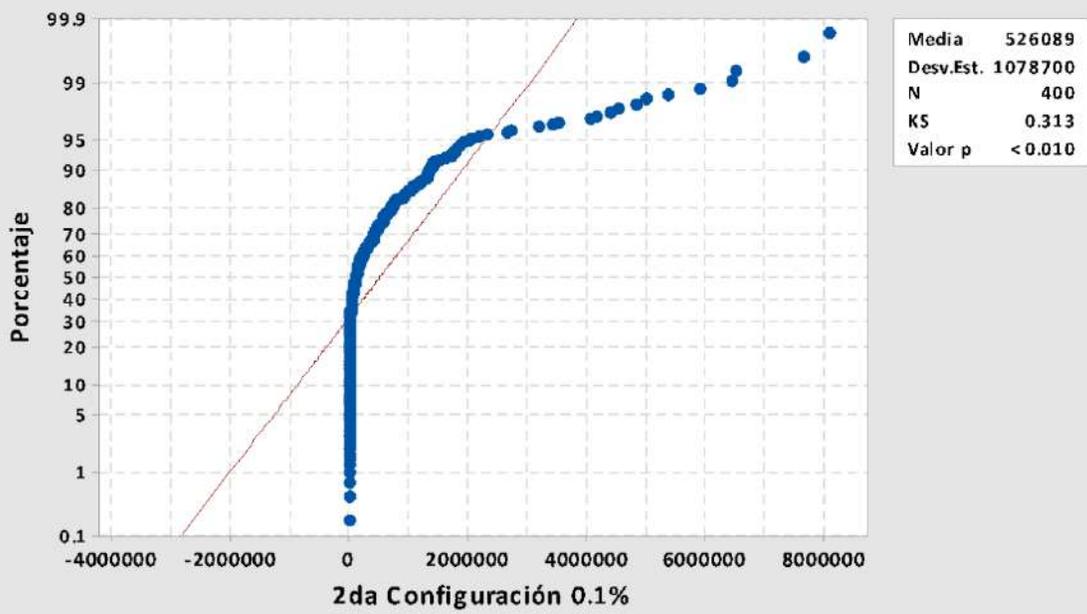
Gráfica de probabilidad de 3ra Configuración 0.3%
Normal



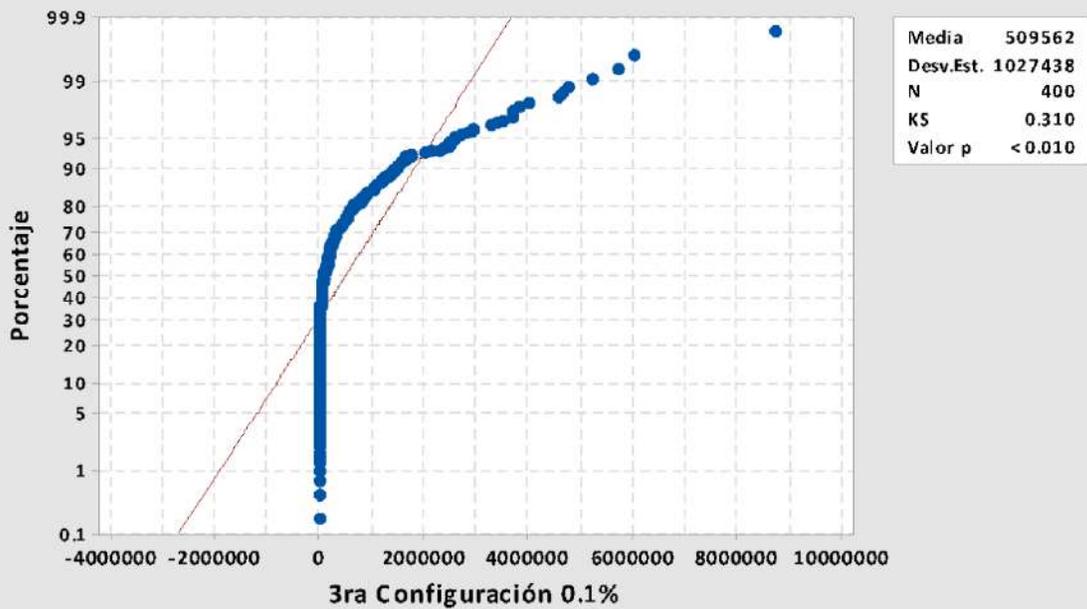
Gráfica de probabilidad de 1ra Configuración 0.1%
Normal



Gráfica de probabilidad de 2da Configuración 0.1%
Normal



Gráfica de probabilidad de 3ra Configuración 0.1%
Normal



Mann-Whitney: 3ra Configuración 1% < 1ra Configuración 1%
 Método

η_1 : mediana de 3ra Configuración 1%

η_2 : mediana de 1ra Configuración 1%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 1%	400	8461.5
1ra Configuración 1%	400	13635.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-1113	-192	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	153383.00	0.018
Ajustado para empates	153383.00	0.018

Mann-Whitney: 3ra Configuración 1% < 2da Configuración 1%
Método

η_1 : mediana de 3ra Configuración 1%

η_2 : mediana de 2da Configuración 1%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 1%	400	8461.5
2da Configuración 1%	400	15210.5

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-785.5	-25	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	154501.50	0.041
Ajustado para empates	154501.50	0.041

Mann-Whitney: 3ra Configuración 0.5% < 1ra Configuración 0.5%
 Método

η_1 : mediana de 3ra Configuración 0.5%

η_2 : mediana de 1ra Configuración 0.5%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.5%	400	14562.5
1ra Configuración 0.5%	400	34745.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-3406	-1169	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	151029.00	0.003
Ajustado para empates	151029.00	0.003

Mann-Whitney: 3ra Configuración 0.5% < 2da Configuración 0.5%
Método

η_1 : mediana de 3ra Configuración 0.5%

η_2 : mediana de 2da Configuración 0.5%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.5%	400	14562.5
2da Configuración 0.5%	400	27259.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-1539	-3	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	154803.50	0.049
Ajustado para empates	154803.50	0.049

Mann-Whitney: 3ra Configuración 0.5% < 1ra Configuración 0.5%
 Método

η_1 : mediana de 3ra Configuración 0.5%

η_2 : mediana de 1ra Configuración 0.5%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.5%	400	14562.5
1ra Configuración 0.5%	400	34745.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-3406	-1169	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	151029.00	0.003
Ajustado para empates	151029.00	0.003

Mann-Whitney: 3ra Configuración 0.3% < 2da Configuración 0.3%
Método

η_1 : mediana de 3ra Configuración 0.3%

η_2 : mediana de 2da Configuración 0.3%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.3%	400	38251.5
2da Configuración 0.3%	400	27250.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
897	3861	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	163033.00	0.807

Mann-Whitney: 3ra Configuración 0.3% < 1ra Configuración 0.3%
 Método

η_1 : mediana de 3ra Configuración 0.3%

η_2 : mediana de 1ra Configuración 0.3%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.3%	400	38251.5
1ra Configuración 0.3%	400	34025.0

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-812	1744	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	158395.50	0.290
Ajustado para empates	158395.50	0.290

Mann-Whitney: 3ra Configuración 0.1% < 2da Configuración 0.1%
 Método

η_1 : mediana de 3ra Configuración 0.1%

η_2 : mediana de 2da Configuración 0.1%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.1%	400	94667
2da Configuración 0.1%	400	117612
Estimación de la diferencia		

Diferencia	Límite superior para la diferencia	Confianza lograda
-3210	2149	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	157413.00	0.197
Ajustado para empates	157413.00	0.197

Mann-Whitney: 3ra Configuración 0.1% < 1ra Configuración 0.1%
Método

η_1 : mediana de 3ra Configuración 0.1%

η_2 : mediana de 1ra Configuración 0.1%

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
3ra Configuración 0.1%	400	94667
1ra Configuración 0.1%	400	123407

Estimación de la diferencia

Diferencia	Límite superior para la diferencia	Confianza lograda
-9478.5	-747	95.00%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 < 0$

Método	Valor W	Valor p
No ajustado para empates	154160.50	0.032
Ajustado para empates	154160.50	0.032

Anexo 5.

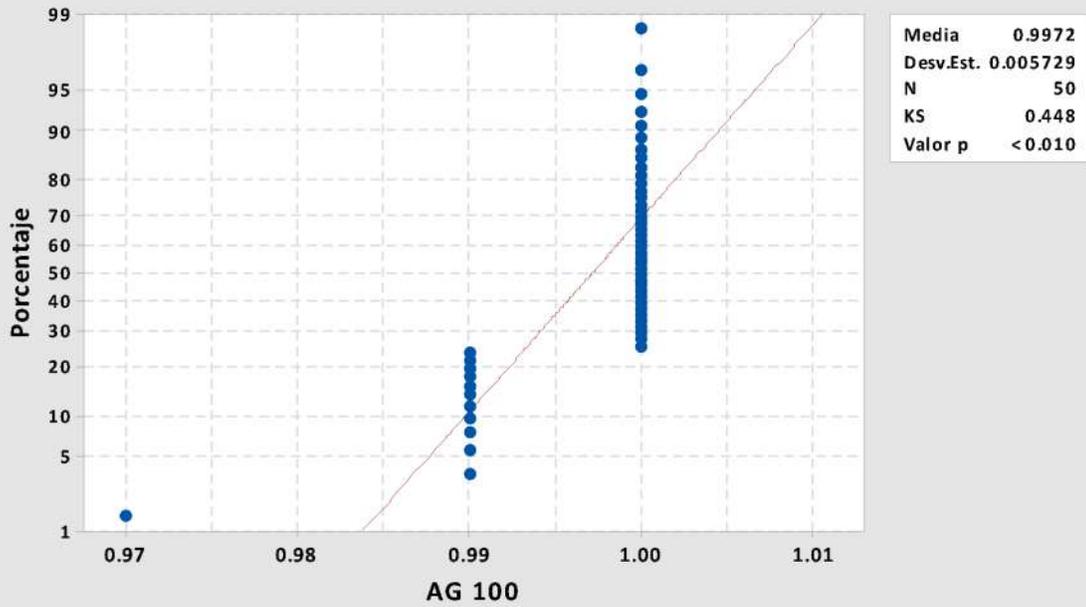
Prueba no paramétrica de Mann-Whitney a los resultados del método PDA implementado con un algoritmo genético y con un algoritmo evolutivo diferencial.

Este anexo contiene el detalle de las pruebas realizadas a los resultados de las tres implementaciones realizadas para el método PDA (vea tabla 3.4). Las gráficas de las páginas siguientes son la prueba de normalidad realizada a cada conjunto de resultados, para revisar que los datos no presentan una distribución normal. Posteriormente, se muestran los resultados obtenidos al realizar la prueba no paramétrica de Mann-Whitney con el software Minitab.

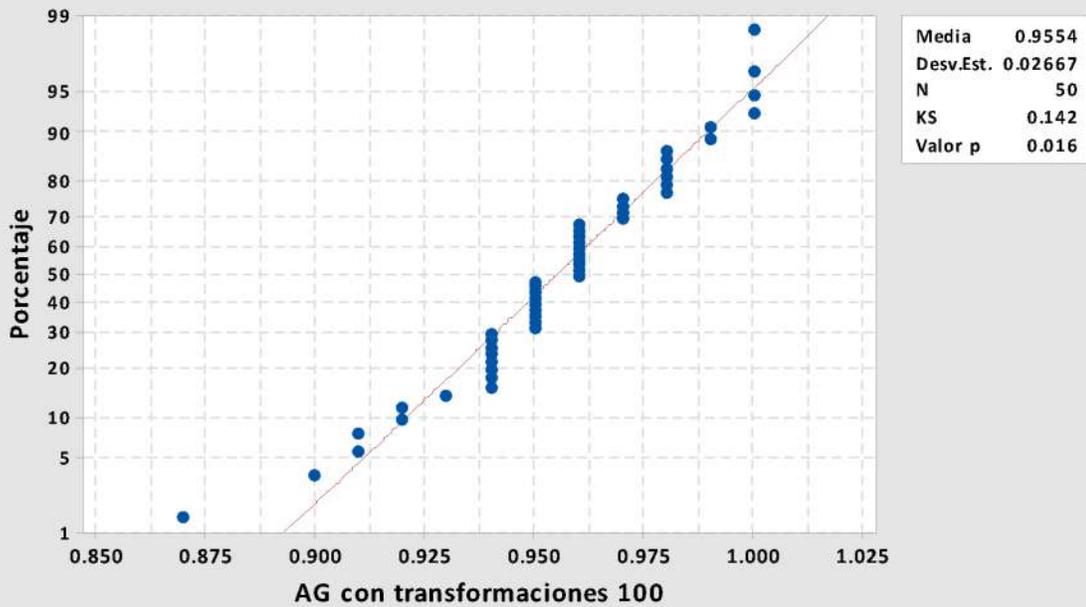
La tabla siguiente muestra las abreviaturas que se utilizarán por sencillez para realizar las pruebas y gráficas en el software Minitab.

Nombre en la tabla 3.4	Nombre utilizado para las pruebas
Algoritmo Genético	AG
Algoritmo Genético con transformaciones	AG con transformaciones
Algoritmo Evolutivo Diferencial con transformaciones	Algoritmo ED

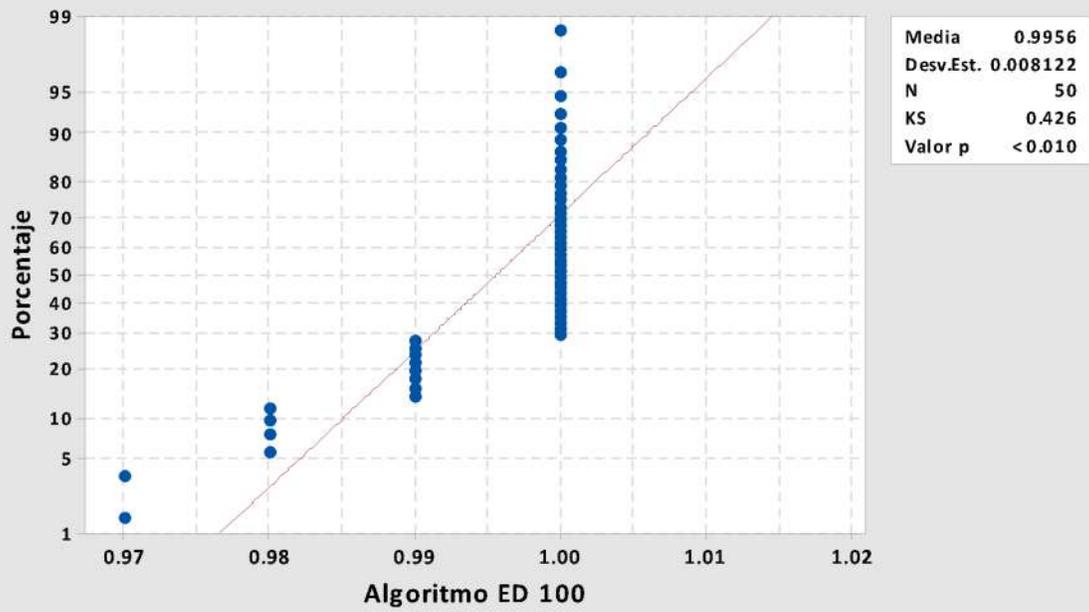
Gráfica de probabilidad de AG 100
Normal



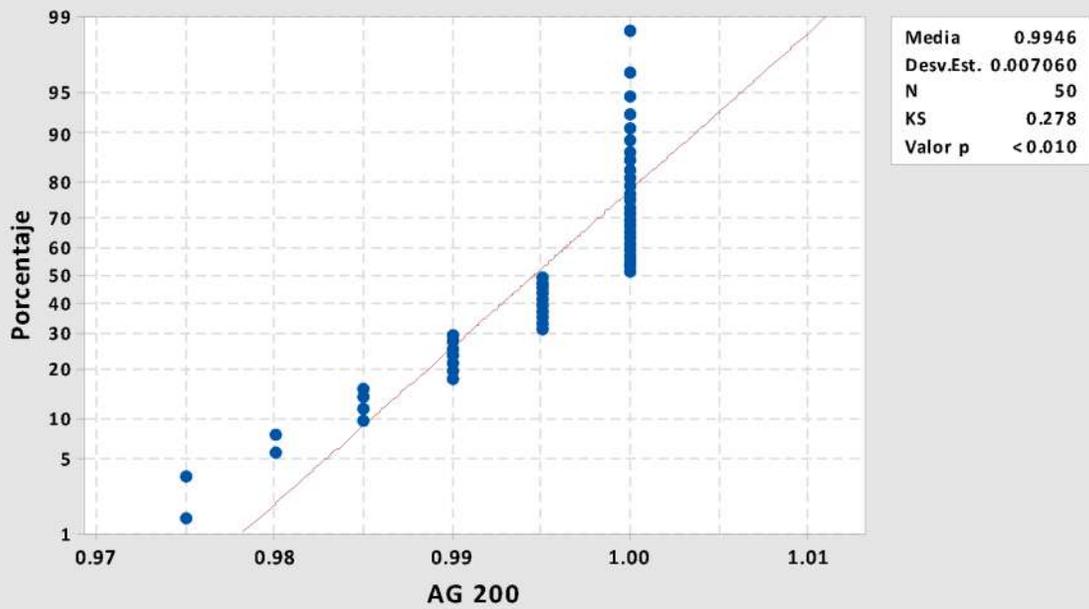
Gráfica de probabilidad de AG con transformaciones 100
Normal



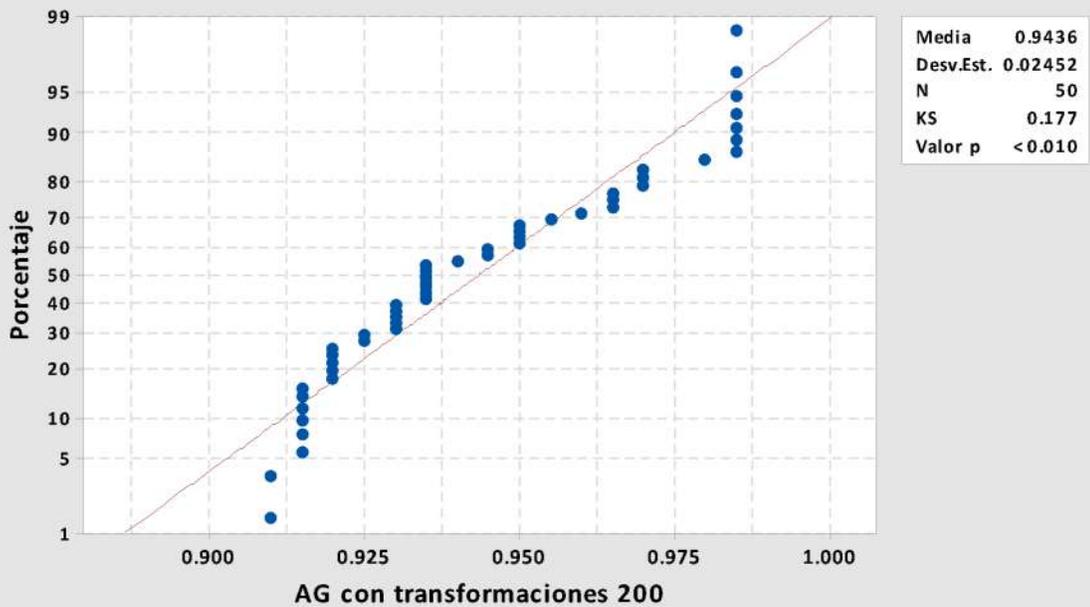
Gráfica de probabilidad de Algoritmo ED 100
Normal



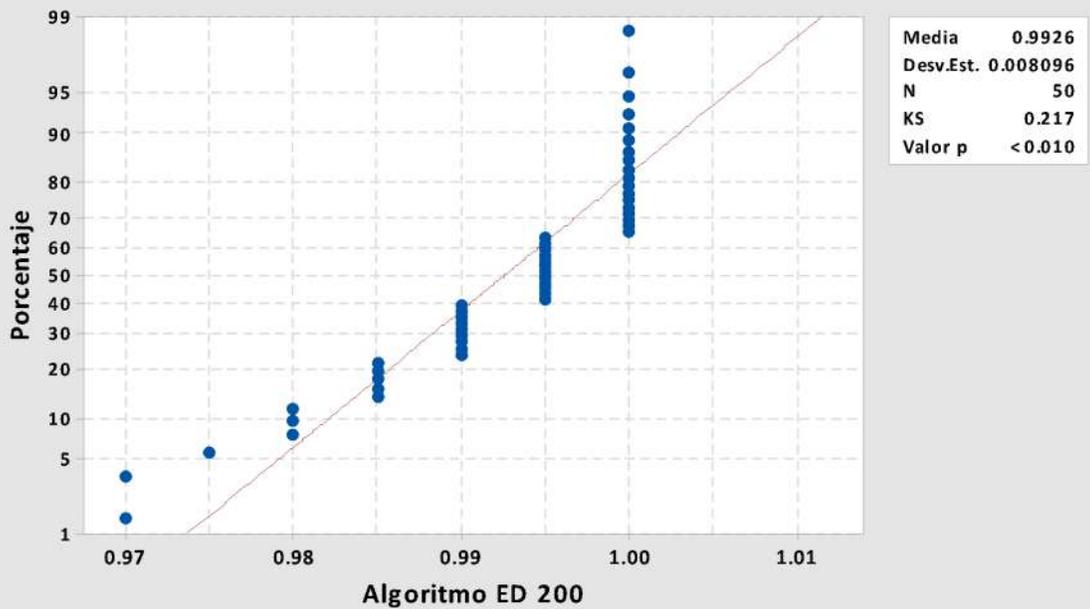
Gráfica de probabilidad de AG 200
Normal



Gráfica de probabilidad de AG con transformaciones 200
Normal



Gráfica de probabilidad de Algoritmo ED 200
Normal



Mann-Whitney: AG 100 > Algoritmo ED 100
Método

η_1 : mediana de AG 100

η_2 : mediana de Algoritmo ED 100

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
AG 100	50	1

Algoritmo ED 100	50	1
------------------	----	---

Estimación de la diferencia

Diferencia	Límite inferior para la diferencia	Confianza lograda
0.00000000	-0.00000000	95.06%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 > 0$

Método	Valor W	Valor p
No ajustado para empates	2602.00	0.299
Ajustado para empates	2602.00	0.246

Mann-Whitney: AG 200 > Algoritmo ED 200
Método

η_1 : mediana de AG 200

η_2 : mediana de Algoritmo ED 200

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
AG 200	50	0.9975
Algoritmo ED 200	50	0.9950

Estimación de la diferencia

Diferencia	Límite inferior para la diferencia	Confianza lograda
0.0000000	-0.0000000	95.06%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 > 0$

Método	Valor W	Valor p
No ajustado para empates	2716.50	0.094
Ajustado para empates	2716.50	0.083

Mann-Whitney: Algoritmo ED 100 > AG 100
Método

η_1 : mediana de Algoritmo ED 100

η_2 : mediana de AG 100

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
Algoritmo ED 100	50	1
AG 100	50	1

Estimación de la diferencia

Diferencia	Límite inferior para la diferencia	Confianza lograda
-0.0000000	-0.0000000	95.06%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 > 0$

Método	Valor W	Valor p
No ajustado para empates	2448.00	0.703

Mann-Whitney: Algoritmo ED 200 > AG 200
Método

η_1 : mediana de Algoritmo ED 200

η_2 : mediana de AG 200

Diferencia: $\eta_1 - \eta_2$

Estadísticas descriptivas

Muestra	N	Mediana
Algoritmo ED 200	50	0.9950
AG 200	50	0.9975

Estimación de la diferencia

Diferencia	Límite inferior para la diferencia	Confianza lograda
-0.0000000	-0.0050000	95.06%

Prueba

Hipótesis nula $H_0: \eta_1 - \eta_2 = 0$

Hipótesis alterna $H_1: \eta_1 - \eta_2 > 0$

Método	Valor W	Valor p
No ajustado para empates	2333.50	0.907

Referencias

Almeida-Dias J., Figueira J., Greco, S., Roy, B. (2010). *“ELECTRE TRI-C: A multiple criteria sorting method based on characteristic reference actions”*. European Journal of Operational Research, Volume 204, Issue 3, Pages 565-580.

Almeida-Dias J., Figueira J., Roy B.(2012). *“A multiple criteria sorting method where eachcategory is characterized by several reference actions: The Electre Tri-nC method”*. European Journalof Operational Research, Elsevier, 2012, 217 (3), pp.567-579.

Blickle, T. and Thiele, L. (1995). *“A comparison of selection schemes used in genetic algorithms”*. Technical Report 11, Computer Engineering and Comunnication Network Lab (TIK), Gloriastrasse 35, 8092 Zurich, Switzerland.

Bouyssou D., Pirlot M., *“On consistent families of criteria: An Axiomatic approach”*. ORBEL 16, 16th Belgian Conference on Quantitative Methods for Decision Making, F.U.S.L., Brussels, January 24–25, 2002.

Bouyssou D., Marchant Th., Pirlot M., Tsoukiàs a., y Vincke, Ph. (2006). *“Evaluation and decision models with multiple criteria: Stepping stones for the analyst”*. Springer Verlag, Boston.

Brans J., Mareschal B. (2016). *“PROMETHEE methods”*. En Multiple Criteria Decision Analysis, State of the Art Surveys. Springer, New York, NY.

Brans J. P. and Mareschal B. (1990). *“The PROMETHEE methods for MCDM; The PROMCALC, GAIA and BANKADVISER software. Readings in MCDA,”*. Bana e Costa, Ed. Berlin: Springer-Verlang, 1990.

Brindle A. (1981). *“Genetic Algorithms for Function Optimization”*. PhD thesis, Department of Computer Science, University of Alberta, Edmonton, Alberta.

Cardinal J.S., Mousseau V., Zheng J.(2011), *“An application of constrained multicriteria sorting to student selection”*. Portfolio Decision Analysis. Volume 162, pp 213-240.

Coello Coello C. A. (1999), *“A comprehensive survey of evolutionary-based multiobjective optimization techniques”*, Knowledge and Information systems. Volume 1, No. 3, , pp 269-308.

Coello C., Lamont, Gary y Van D. (2002). *“Evolutionary Algorithms for Solving Multi-Objective Problems”*. Genetic and Evolutionary Computation. USA: Springer.

Coello C. and Lamont G. (2004). *“Applications of Multi-Objective Evolutionary Algorithms”*. USA: World Scientific.

Coello C. A., Van Veldhuizen, D. A. y Lamont, G. B. (2007). *“Evolutionary Algorithms for Solving Multi-Objective Problems”*. Second Edition. Springer, New York.

Coello Coello C. A. (2018), *“Introducción a la Computación Evolutiva”*, CINVESTAV-IPN, Mayo 2018.

Covantes E., Fernandez E., Navarro J. (2016). *“A combined PDA-Theseus framework for multi-criteria sorting”*, Foundations of Computing and Decision Sciences, Vol. 41, pp. 213-235.

Covantes E., Fernández E., Navarro J. (2016). *“Handling the multiplicity of solutions in a MOEA-based PDA-THESEUS framework for multi-criteria sorting”*, Foundations of Computing and Decision Sciences vol. 41, no. 4, 2016, pp. 213-235.
DOI: <http://dx.doi.org/10.1515/fcds-2016-0013>

Cruz L., Fernández E., Rangel N. (2017). *“A metaheuristic optimization-based indirect elicitation of preference parameters for solving many-objective problems ”*, International Journal of Computational Intelligence Systems vol. 10, no. 1, 2017, pp. 56-77.

Deb K., Pratap A., Agarwal S. and Meyarivan T. (2002), *“A fast and elitist multiobjective genetic algorithm: NSGA-II”*, in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197.

DeJong A.K. (1975). *“An Analysis of the Behavior of a Class of Genetic Adaptive Systems”*. PhD thesis, University of Michigan.

Devaud J.M., Groussaud G. and Jacquet-Lagrèze E. (1980). *“UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux”*. European working group on MCDA, Bochum , Germany, 1980.

Dias, L., Mousseau, V., Figueira J., Climaco, J. (2002). *“An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE-TRI”*, European Journal of Operational Research, 138, 2, 332-348.

Dias, L., Mousseau, V. (2006). *“Inferring ELECTRE's veto-related parameters from outranking examples,”*, European Journal of Operational Research, 170, 1, 172-191.

Doumpos, M. y Zopounidis, C. (2004). *“Multicriteria Decision Aid Classification Methods”*. En Pardalos, P. and Hearn, D. (Eds.). Multicriteria Decision Aid Classifications Methods. USA. Kluwer Academic Publishers.

Doumpos M., Marinakis Y., Marinaki M., Zopounidis C. (2009), *“An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method”*. European Journal of Operational Research. Vol.199, 496–505.

Doumpos, M. and Zopoundis C. (2011), ***“Preference disaggregation and statistical learning for multicriteria decision support: A review”***. European Journal of Operational Research. Vol. 209, pp. 203–214.

Edwards W. and Barron F.H. (1994), ***“SMARTS and SMARTER: improves simple methods for multiattribute utility measurement”***, Organizational Behaviour and Human Decision Processes, vol. 60, no. 1, pp. 306-25.

Environmental Protection Agency. (1992) ***“Facility Pollution Prevention Guide”***, Office of Research and Development Environmental Protection Agency, Washington.

Fernández, E., Leyva, J., 2004. ***“A method based on multiobjective optimization for deriving a ranking from a fuzzy preference relation”***. En European Journal of Operational Research 154, 110–124.

Fernandez, E., Cancela, N., Olmedo, R. (2008). ***“Deriving a final ranking from fuzzy preferences: An approach compatible with the Principle of Correspondence”***. Mathematical and Computer Modelling, Vol 47. pp 218–234.

Fernandez, E., Navarro, J., Bernal, S. (2009): ***“Multicriteria sorting using a valued indifference relation under a preference disaggregation paradigm”***, European Journal of Operational Research, 198, 2, 602-609.

Fernandez E., Navarro J., Mazcorro G.(2012): ***“Evolutionary multi-objective optimization for inferring outranking model’s parameters under scarce reference information and effects of reinforced preference”***, Foundations of Computing and Decision Sciences, 37, 3, 163-197.

Fernandez E., Navarro J., Salomon E. (2014): ***“Automatic enhancement of the reference set for multi-criteria sorting in the frame of THESEUS method”***, Foundations of Computing and Decision Sciences 39, 2, 57-77.

Fernandez, E., Figueira, J., Navarro, J., Roy, B. (2017): ***“ELECTRE TRI-nB: A new multiple criteria ordinal classification method”***, European Journal of Operational Research, 263, 2, 214-224.

Fernandez, E., Gaxiola, O., Navarro, J. (2018): ***“Un nuevo enfoque para resolver el problema de clasificación ordinal con restricciones al número de objetos en las categorías”***, Revista Investigación Operacional. Forthcoming Paper #90B50-18-08. <https://rev-inv-ope.univ-paris1.fr/volumes-since-2000/forthcoming/>.

Figueira, J., Mousseau, V., Roy, B. (2005): ***“Electre Methods”*** en Figueira, J., Greco, S. y Erghott, L. (eds.) Multiple Criteria Decision Analysis : State of the art Surveys, Springer Science+Business Media, New York.

Figueira J., Greco, S., Roy, B. y Slowinski, R. (2010). ***ELECTRE Methods: Main Features and Recent Developments***. En Constantin Zopounidis y Pardalos Panos (Eds.). Handbook in Multicriteria Analysis. Berlin, Alemania. Springer-Verlag.

Figueira J.R., Mousseau V., Roy B. (2016). ***ELECTRE Methods***. In: Greco S., Ehrgott M., Figueira J. (eds) Multiple Criteria Decision Analysis. International Series in Operations Research & Management Science, vol 233. Pp 155-185. Springer, New York, NY.

Fishburn P. (1970). ***Utility theory for decision making***, Wiley.

Fodor, J. y Roubens, M. (1992). ***Aggregation of strict preference relations in MCDM procedures***, in: V. Novák et al. (Eds.), Fuzzy Approach and Decision-Making, (Academia, Praha) : pp. 163-171.

Fonseca, C.M. y Fleming, P.J. (1993). ***Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization***. En Genetic Algorithms: Proceedings of the Fifth International Conference, 416-423. Morgan Kaufmann.

Fontana M. and Cavalcante C. (2013). ***ELECTRE Tri method used to storage location assignment into categories***. Brazilian Operations Research Society 33(2): 283-303 .

French S. (1986). ***Decision Theory: An Introduction to the Mathematics of Rationality***, Ellis Horwood series, University of London.

Garcia, M. S. (2009): ***Métodos para la comparación de alternativas mediante un Sistema de Ayuda a la Decisión (S.A.D.) y Soft Computing***, Tesis de Doctorado, Universidad Politécnica de Cartagena.

Gastélum Chavira, Diego .A. (2014). ***Un enfoque evolutivo multiobjetivo para el problema de ordenamiento multicriterio de un conjunto de alternativas de tamaño mediano***. Tesis Doctoral, Universidad de Occidente.

Goldberg David E.(1989). ***Genetic Algorithms in Search, Optimization and Machine Learning***. Addison-Wesley Publishing Co., Reading, Massachusetts.

Gomes H., Costa A. & Fernandes P. (2004). ***The ELECTRE TRI method applied to the evaluation of companies transporting hazardous materials***. Gestão & Produção, 11(2), 221-229.

Gomes H., Uébe A., Policani A., De Carvalho R (2007). ***ELECTRE TRI applied to evaluation of consumer satisfaction***. Produção, v. 17, n. 2, p. 230-245.

Gunter R. (1994). ***Convergence Analysis of Canonical Genetic Algorithms***. IEEE Transactions on Neural Networks, 5:96–101, January 1994.

Kablan, M.M. (2004). “**Decision Support for Energy Conservation Promotion: An Analytic Hierarchy Process Approach**”. Energy Policy, 32, 1151-1158. [http://dx.doi.org/10.1016/S0301-4215\(03\)00078-8](http://dx.doi.org/10.1016/S0301-4215(03)00078-8)

Kadzinski M. and Slowinski R. (2013), “**DIS-CARD: a new method of multiple criteria sorting to classes with desired cardinality**”. Journal of Global Optimization. Volume 56, Issue 3 , pp 1143-1166.

Kadzinski M. and Slowinski R. (2015), “**Parametric evaluation of research units with respect to reference profiles**”. Decision Support Systems. Vol 72. 2015, pp 33-43.

Keeney R. and H. Raiffa. (1976). “**Decisions with Multiple Objectives: Preferences and Values Tradeoffs**”, Wiley, New York ed 1976.

Köksalan M., Mousseau V. and Özpeynirci S. (2017), “**Multi-Criteria Sorting with Category Size Restrictions**”. International Journal of Information Technology & Decision Making Vol. 16, No. 1 , pp. 5–23.

Lu G., Wang H. and Mao X. (2010). “**Using ELECTRE TRI Outranking Method to Evaluate Trustworthy Software**”. Autonomic and Trusted Computing pp 219-227.

Mailly, D., Abi-Zeid, I. And Pepin, S. (2013). “**Multi-Criteria Classification Approach for Identifying Favourable Climates for Tourism**”. Journal of Multi-Criteria Decision Analysis vol. 21, 65-75.

Mousseau, V., Slowinski, R.(1998): “**Inferring an ELECTRE-TRI model from assignment examples**”, Journal of Global Optimization, 12, 2, 157-174.

Mousseau, V., Slowinski, R., Zielniewicz, P. (2000). “**A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support**”. Computers & Operations Research, No. 27, 757-777.

Mousseau, V., Figueira, J., Dias, L. (2003), “**On the Notion of Category Size in Multiple Criteria Sorting Models**”. DIMACS Technical Report 2003-02 ,February 2003.

Norese, M. F. and Carbone, V. (2014), “**An Application of ELECTRE Tri to Support Innovation**”. J. Multi-Crit. Decis. Anal., 21: 77–93.

Olmedo Pérez Rafael A. (2009). “**Avances en la modelación basada en relaciones borrosas de sobreclasificación para sistemas de apoyo a la decisión en grupo**”. Tesis doctoral. Universidad Autónoma de Sinaloa.

Ostanello, A., (1983): “**Outranking Methods**”, In B. Fandel, G., Spronk, J., Matarazzo, Eds., International Summer School on Multiple Criteria Decision Making Methods, Applications and Software 1, Acireale, Italy, pp. 41-60.

Özpeynirci, S., Özpeynirci, Ö. & Mousseau, V. (2017), “*An interactive algorithm for multiple criteria constrained sorting problem*”. Annals of Operations Research, 2017. On line: <https://doi.org/10.1007/s10479-017-2418-2>.

Perny P. (1998), “*Multicriteria filtering methods based on concordance and non-discordance principles*”. Annals of Operations Research, 80:137–165.

Price, K., Storn, R., Lampinen, J., (2005). “*Differential Evolution: A Practical Approach to Global Optimization*”. Springer, Heidelberg.

Roy, B. (1985). “*Méthodologie Multicritère D'aide à la Décision*”. En Economica. Paris.

Roy B, (1991). “*Decision-Aid and Decision Making*”, European Journal of Operational Research, vol. 45, pp. 324-331.

Siarry P, Michalewicz Z. (2008). “*Advances in metaheuristics for hard optimization*”. New York: Springer.

Saaty, T.L. (1980). “*The Analytic Hierarchy Process*”. McGraw-Hill, New York.

Storn, R., (1996). “*On the usage of differential evolution for function optimization*”. In: Smith, M., Lee, M., Keller, J., Yen, J. (Eds.), NAFIPS 1996, Berkley. IEEE Press, New York, pp. 519–523.

Storn R., Price K. (1997), “*Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*”. Journal of Global Optimization 11. 341-359.

Vinke J. P. and Brans J. P., “*A preference ranking organization method (the PROMETHEE method for multiple criteria decision-making)*”, Management science, vol. 31, no. 6, pp. 647-656, 1985.

Von Neumann J., Morgenstern O.,(1944) “*Theory of games and economicbehaviour*”. Princeton: Princeton University Press, 1944.

Wetzel A. (1983). “*Evaluation of the effectiveness of genetic algorithms in combinatorial optimization*”. PhD thesis, University of Pittsburgh, Pittsburgh, Philadelphia, USA.

Zhang Q., Li H. (2007), “*MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition*”. Ieee Transactions On Evolutionary Computation, Vol. 11, No. 6,pp 712-731, December 2007.

Zheng J., Cailloux O., Mousseau V. (2011), “*Constrained multicriteria sorting method applied to portfolio selection*”. Algorithmic Decision Theory. Volume 6992, pp 331-343.

Zitzler E., Laumanns M. and Thiele L. (2001). ***"SPEA2: Improving the Strength Pareto Evolutionary Algorithm"***. TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland. 2001.

Zopounidis, C. y Pardalos P. (Eds.) (2010). ***"Handbook of Multicriteria Analysis"***. Alemania: Springer Verlag.