



**TECNOLÓGICO
NACIONAL DE MÉXICO**

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

División de Estudios de Posgrado e Investigación

Maestría en Ciencias de la Computación



TESIS

**DESARROLLO DE ALGORITMO PARALELO BASADO EN PREFERENCIAS PARA LA
SELECCIÓN DE CARTERA PROYECTOS**

**Que para obtener el Grado de
Maestro en Ciencias de la Computación**

**Presenta
Ing. Juan Carlos Hernández Marín
G09070369**

**Director de Tesis
Dr. Nelson Rangel Valdez**

**Co-Director de Tesis
Dra. Claudia Guadalupe Gómez Santillán**



Cd. Madero, Tams., a **27 de Noviembre de 2018**

OFICIO No. US.093/18
ÁREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS.

ING. JUAN CARLOS HERNÁNDEZ MARÍN
No. DE CONTROL G 09070369
PRESENTE

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su Examen de Grado de Maestro en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

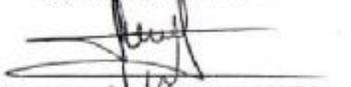
"DESARROLLO DE ALGORITMO PARALELO BASADO EN PREFERENCIAS PARA LA SELECCIÓN DE CARTERA DE PROYECTOS"

El Jurado está integrado por los siguientes catedráticos:

PRESIDENTE :	DR.	JUAN FRAUSTRO SOLÍS
SECRETARIO:	DRA.	LAURA CRUZ REYES
VOCAL:	DR.	NELSON RANGEL VALDEZ
SUPLENTE:	DRA.	CLAUDIA GUADALUPE GÓMEZ SANTILLÁN
DIRECTOR DE TESIS :	DR.	NELSON RANGEL VALDEZ
CO-DIRECTORA DE TESIS:	DRA.	CLAUDIA GUADALUPE GÓMEZ SANTILLÁN

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE
*Excelencia en Educación Tecnológica**
"POR MI PATRIA Y POR MI BIEN"


DR. JOSÉ AARÓN MELO BANDA
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN


SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO
E INVESTIGACIÓN

c.c.p.- Archivo
Minuta

JAMB \JAMF \mdeca*



Declaración de Originalidad

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Cd. Madero y sus autoridades.

7 de Diciembre de 2018, Cd. Madero, Tamps.



Ing. Juan Carlos Hernández Marín

Agradecimientos.

Expreso mis agradecimientos a todas las personas que influyeron de manera directa o indirecta para la realización de este trabajo, principalmente a mis directores Dr. Nelson Rangel Valdez ya que me brindó su apoyo incondicional y a la Dra. Claudia Guadalupe Gómez Santillán que lejos de brindarme comentarios enriquecedores para este.

Además le agradezco a mis compañeros por brindarme su amistad y apoyo incondicional, así como también le agradezco a mis maestros del posgrado en Ciencias de la Computaciones. También le agradezco de manera muy especial a la familia del Ángel Franco por su apoyo incondicional.

Mi más profundo agradecimiento al Consejo Nacional de Ciencia y Tecnología (CONACYT), y al Instituto Tecnológico de Ciudad Madero (ITCM), por haberme otorgado el apoyo en la realización de este proyecto.

Contenido

1. Introducción.....	1
1.1. Motivaciones.....	2
1.2. Justificación.....	3
1.3. Objetivos.....	4
1.3.1. Objetivo General.....	4
1.3.2. Objetivos Específicos.....	4
1.4. Alcances y limitaciones.....	4
2. Marco Teórico.....	5
2.1. Problema de Optimización.....	5
2.2. Problemas de Optimización Multi Criterio.....	6
2.3. Cartera de Proyectos.....	7
2.4. Cartera de Proyectos Públicos.....	8
2.5. Meta-Heurísticas.....	10
2.6. Algoritmos Genéticos.....	10
2.7. Algoritmos de Optimización por Colonia de Hormigas.....	11
2.7.1. Función de Aptitud para Problemas Multi Objetivo.....	13
2.7.2. Dominancia.....	13
2.7.3. Fast Non-Dominated Sorting.....	13
2.7.4. Estimación de la Densidad.....	14
2.8. Toma de Decisiones.....	15
2.8.1. Modelos de Preferencia.....	16
2.8.2. Métodos ELECTRE.....	17
2.8.3. Modelo de Sobre Clasificación Difusa.....	19
2.9. Estrategias de Paralelización para Meta heurísticas.....	20
2.9.1. Maestro – Esclavo (paralelización global).....	21
2.9.2. Grano Grueso.....	21
2.9.3. Grano Fino.....	21
2.9.4. Grano Grueso y Fino.....	22
2.9.5. Grano Grueso y Maestro – Esclavo.....	22
2.9.6. Grano Grueso a Dos Niveles.....	23
2.9.7. Paralelismo por Tareas.....	23

2.9.8.	Divide y vencerás.....	23
2.10.	Aceleración o Speedup, estrategia para medir el desempeño de un Algoritmo Paralelo.	23
3.	Trabajos Relacionados.....	24
3.1.	Manejo de Preferencias.	25
3.2.	Algoritmos paralelos.	26
3.3.	Algoritmos que incorporen preferencias para la solución del problema CPP.	27
3.4.	Conclusiones de la revisión de la literatura científica.....	27
4.	Descripción del problema.....	28
4.1.	Generalidades del Problema de cartera de Proyectos.....	29
4.2.	Incorporación de Preferencias.....	29
4.3.	La mejor Cartera.....	30
5.	Metodología de Desarrollo.	32
6.	Propuesta de solución.....	33
6.1.	Implementación de Algoritmos Meta heurísticos.....	33
6.1.1.	Algoritmos Genéticos.....	33
6.1.2.	Algoritmo NSGA-II.	33
6.1.3.	Algoritmo NOSGA-II.....	35
6.1.4.	Características de los GA's implementados.	35
6.1.5.	Algoritmo de Optimización por Colonia de Hormigas.....	36
6.1.6.	Características de los algoritmos ACO.....	37
6.2.	Características de la implementación de los Algoritmos paralelos.	37
7.	Validación Experimental.....	38
7.1.	Análisis de los Algoritmos Meta heurísticos secuenciales.	39
7.2.	Resultados derivados de la Experimentación.	42
7.3.	Descripción de la aplicabilidad de los patrones de diseño al algoritmo NOACO.....	57
7.3.1.	Implementación de los algoritmos basados en OpenMP.	57
7.3.2.	Implementación de los algoritmos basados en MPI.	60
7.3.3.	Implementación del Algoritmo Híbrido.	62
7.4.	Análisis de los algoritmos Paralelos.	63
7.4.1.	Análisis del Algoritmo OMP-FD.	63
7.4.2.	Análisis del algoritmo OMP-GG.....	68
7.4.3.	Análisis del Algoritmo MPI-MEGG.....	73

7.4.4.	Análisis del Algoritmo MPI-MEFD.	79
7.4.5.	Comentarios derivados de la experimentación de los Algoritmos implementados.	83
7.4.6.	Análisis del algoritmo MEGGFD.....	87
8.	Conclusiones.....	92
9.	Referencias.....	95
	Índice de Figuras	100
	Índice Tabla	100

Índice de Figuras

Figura 1.-	Clasificación de técnicas EMOO(Victoria Ruiz Martínez, 2016).....	7
Figura 2.-	Algoritmo fast non dominated sorting.....	14
Figura 3.-	Distancia de Crowding.	15
Figura 4.-	Estrategia Maestro Esclavo.	21
Figura 5.-	Estrategia de Grano Grueso	21
Figura 6.-	Estrategia de Grano Fino.....	22
Figura 7.-	Estrategia de Grano Grueso y Fino.	22
Figura 8.-	Estrategia de Maestro-Esclavo y Grano Grueso.....	22
Figura 9.-	Estrategia de Grano Grueso a Dos Niveles.....	23
Figura 10.-	Metodología de desarrollo.....	32
Figura 11.-	Algoritmo NSGA-II.	34
Figura 12.-	Algoritmo NOSGA II.....	35
Figura 13.-	Algoritmo ACO.	36
Figura 14.-	Procedimiento para evaluación del frente no Dominado.....	40
Figura 15.-	Calculo del frente no dominado.....	40
Figura 16.-	Análisis del Modelo de Sobre-clasificación.....	41
Figura 17.-	Esquema de Grano Grueso.	58
Figura 18.-	Esquema Flujo de Datos.....	59
Figura 19.	Esquema del Algoritmo MPI-MEGG.	60
Figura 20.-	Diagrama MPI-FDME.....	61
Figura 21.-	Diagrama del Diseño H-MEGGFD.....	62

Índice Tabla

Tabla 2.-	Análisis de estado del arte.	28
Tabla 3.-	Comportamiento de individuos en el no dominado del algoritmo NS-ACO.	42
Tabla 4.-	Comportamiento de los frentes del algoritmo NS-ACO.	43
Tabla 5.-	Comportamiento de los individuos en el no dominado del algoritmo NSGA-II.....	45
Tabla 6.-	Comportamiento de los frentes del algoritmo NSGA-II	46
Tabla 7.-	Comportamiento de individuos en el no dominado del algoritmo NOSGA-II.....	47

Tabla 8.- Comportamiento de los frentes generados por el algoritmo NOSGA-II	47
Tabla 9.- Comportamiento de las soluciones en el frente no dominado del algoritmo NO-ACO....	48
Tabla 10.- Comportamiento de los frentes generados por el algoritmo NO-ACO.....	49
Tabla 11.- Comparación del frente no dominado de los algoritmos Genéticos en el indicador Min y Max.....	50
Tabla 12.-Comparación del frente no dominado de los algoritmos Genéticos en el indicador Promedio y Varianza.	51
Tabla 13.-Comparación del frente no dominado de los algoritmos ACO en el indicador Min y Max.	52
Tabla 14.-Comparación del frente no dominado de los algoritmos ACO en el indicador Promedio y Varianza.....	53
Tabla 15.- Análisis del frente no dominado de los algoritmo NSGA-II y NOSGA-II.	54
Tabla 16.- Análisis del frente no dominado de los algoritmo NS-ACO y NO-ACO.....	55
Tabla 17.- Análisis de la Frontera no Superada Estrictamente.	56
Tabla 18.- Incidencia de General del frente no dominado del Algoritmo OMP-GG.	64
Tabla 19.- Indicador de Aceleración Mínima del algoritmo OMP-FD.	65
Tabla 20.- Indicador de Aceleración Máxima del algoritmo OMP-FD.....	66
Tabla 21.- Indicador de Aceleración Promedio del algoritmo OMP-FD.	67
Tabla 22.- Indicador de la Varianza obtenida por la Aceleración del algoritmo OMP-FD.	68
Tabla 23.- Frente No Dominado del algoritmo OMP-GG	69
Tabla 24.-Indicador Aceleración Mínima el algoritmo OMP-GG.....	70
Tabla 25.- Indicador Aceleración Máxima el algoritmo OMP-GG.	71
Tabla 26.- Indicador de Aceleración Promedio del algoritmo OMP-GG.	72
Tabla 27.- Indicador de la varianza de la Aceleración del algoritmo OMP-GG.	73
Tabla 28.-Frente No Dominado del algoritmo MPI-MEGG.	74
Tabla 29.- Incidencia de soluciones del Frente No Dominado del algoritmo MPI-MEGG.	75
Tabla 30.- Indicador Aceleración Mínima el algoritmo MPI-MEG.	76
Tabla 31.- Indicador Aceleración Máxima el algoritmo MPI-MEGG.	76
Tabla 32.-Indicador de Aceleración del algoritmo MPI-MEGG.	77
Tabla 33.- Indicador de la varianza obtenida de la Aceleración del algoritmo MPI-MEGG.	78
Tabla 34.- Frente No Dominado del algoritmo MPI-MEFD.	79
Tabla 35.- Indicador Aceleración Mínima del algoritmo MPI-MEFD.....	80
Tabla 36.- Indicador Aceleración Máxima del algoritmo MPI-MEFD.....	81
Tabla 37.- Indicador Aceleración Promedio del algoritmo MPI-MEFD.	82
Tabla 38.- Indicador de la varianza de la Aceleración del algoritmo MPI-MEFD.	83
Tabla 39.- Experimentaciones que superaron en calidad de solución a la versión secuencial.....	83
Tabla 40.- Ranking obtenido por la prueba de Friedman.	86
Tabla 41.- prueba post-hoc de Bonferroni-Dunn.	87
Tabla 42.- Frente no dominado del algoritmo híbrido.....	88
Tabla 43.- Indicador de Aceleración Mínima del algoritmo Híbrido.....	89
Tabla 44.-Indicador de Aceleración Máxima del algoritmo Híbrido.	90
Tabla 45.- Indicador de Aceleración Promedio del algoritmo Híbrido.....	91
Tabla 46.- Indicador de la varianza obtenida de la Aceleración del algoritmo Híbrido.	92

1. Introducción.

Esta tesis tiene como principales aportaciones el solucionar el problema de optimización multi-objetivo conocido como Carrera de Proyectos en caso particular del financiamiento de proyectos públicos, como resultado de la solución de este problema se obtiene un conjunto de posibles alternativas de solución y cada una de estas alternativas indica cómo será distribuido el presupuesto.

Posteriormente de obtener las soluciones es necesario hacer un análisis exhaustivo para determinar cuál es la solución que presenta un mejor escenario ya que solo se puede financiar una cartera, en este análisis la responsabilidad recae en el Tomador de decisiones (DM por sus siglas en inglés Decision Marker) ya que es el que decide cual es la cartera que será financiada.

Otra de las aportaciones de esta tesis es incorporar la información necesaria al algoritmo de solución para simular el proceso de comparación de alternativas de solución y así disminuir la carga cognitiva del DM, este proceso es denominado incorporación de preferencias.

Ahora bien, los factores que pueden afectar a un algoritmo en su desempeño, en la aplicación de este problema, podrían ser: a) el tamaño del problema; b) el número de objetivos involucrados; y c) estrategias algorítmicas utilizadas. De particular interés se cita al empleo de los enfoques secuenciales. Si bien, a la fecha existen muchos algoritmos secuenciales que han resuelto el problema de cartera de proyectos de manera eficiente, con y sin incorporación de preferencias, se resalta la necesidad de extender el área de conocimiento hacia la inclusión de diseños paralelos, principalmente, porque de acuerdo a lo que se observa en la literatura científica, estos ayudan a reducir los tiempos de solución, y en el caso particular de estrategias que incorporen preferencias para la solución del problema de cartera de proyectos públicos, estos son relativamente escasos.

Por lo que el objetivo de esta tesis es implementar meta heurísticas con y sin incorporación de preferencias los algoritmos implementados fueron NSGA-II, NOSGAI, NSACO y NOACO los detalles de cada una de las experimentaciones de muestran se describen en este documento. Una de las actividades de este proyecto de tesis fue identificar cual es el algoritmo más prometedor en términos de la calidad de solución y tiempo de ejecución.

A lo largo de este capítulo se dan a conocer cada uno de los elementos que justifican la realización de este proyecto de tesis. Los aspectos abordados en este capítulo son: a) los objetivos que se alcanzaron en esta investigación; así como también b) La justificación del proyecto de investigación, c) Antecedentes y finalmente d) los alcances y limitaciones del proyecto de tesis.

En el capítulo 2 de este documento se dan a conocer los conceptos introductorios de las áreas de conocimiento asociadas a las problemáticas resueltas. Una de las actividades realizadas en este trabajo de investigación fue la revisión de la literatura científica para posteriormente se realizó un análisis de la información recabada.

Con el análisis de la literatura científica se encontraron trabajos relacionados con las áreas de impacto del proyecto de investigación de tesis, los cuales se muestran en el capítulo 3 de este documento.

Posteriormente en el capítulo 4 se describen cada una de las problemáticas resueltas en este proyecto de investigación donde se destaca el impacto de la contribución de este proyecto. En el siguiente capítulo se describe cada una de las actividades realizadas en la metodología de trabajo para el desarrollo de este proyecto de investigación.

En el capítulo 6 se describe la propuesta de solución de este proyecto de investigación, en el cual se describe el funcionamiento de cada una de las meta heurísticas implementadas y además se describen las características particulares de cada uno de los algoritmos.

En el siguiente capítulo se describen cada una de las experimentaciones resueltas en este proyecto de investigación. Las experimentaciones realizadas tienen como finalidad identificar los algoritmos que tienen una buena calidad de solución y a su vez reporten buenos tiempos de ejecución.

La primera experimentación se realizó con la finalidad de identificar el mejor algoritmo secuencial en términos de la calidad de solución, una vez identificado se propusieron patrones de diseño para paralelizar esta meta heurística. Además en este capítulo se describen las implementaciones de cada uno de los patrones de diseño utilizados para la implementación de las meta heurísticas.

Además en este capítulo también se describe las experimentaciones realizadas para determinar los beneficios de cada uno de los algoritmos paralelos implementados con la finalidad de identificar el entorno de ejecución en el cual los algoritmos presentan buenos resultados en términos de la calidad de solución y reducción del tiempo de ejecución. Finalmente en este capítulo se dan a conocer las conclusiones derivadas de las experimentaciones realizadas.

Finalmente en el capítulo 8 dan a conocer las conclusiones derivadas de este trabajo de tesis, entre las cuales destacan, la identificación del mejor algoritmo secuencial, el algoritmo paralelo que obtuvo el mejor comportamiento, así como también la aplicabilidad de los patrones de diseño.

1.1. Motivaciones.

El resolver un problema de combinatoria de alta complejidad representa un reto no solo al ámbito científico sino que también, para el sector productivo con la finalidad de mejorar ganancias, disminuir tiempos de entrega, tener una buena distribución entre otras.

Este reto se debe principalmente se debe al costo computacional que se necesita para poder encontrar la mejor o las mejores alternativas de solución para la empresa, ya que este costo incrementa de manera exponencial dependiendo de la magnitud del problema es decir el número de variables que son involucradas para la solución al problema.

En este proyecto de investigación se resuelve el problema de Cartera de Proyectos Públicos (CPP) el cual tiene como finalidad determinar cuál es el subconjunto de proyectos que pueden ser financiados con un presupuesto definido, y que lleve consigo los mejores beneficios a la sociedad.

Además el problema de administración de recursos no solo es característico para resolver las problemáticas de carácter social sino también en el sector productivo, ya que la distribución de los recursos determina el éxito o fracaso de la organización.

Otro de los aspectos más sobresalientes para la solución de este problema es la optimización de múltiples objetivos de manera simultánea lo cual trae consigo el problema de dimensionalidad lo cual hace que el espacio de búsqueda de soluciones crezca de manera exponencial, por lo cual incrementa la complejidad.

Debido a la complejidad que con lleva obtener la mejor o mejores alternativas de solución alternativas mediante un algoritmo exacto se propone darle solución al problema mediante el uso de meta heurísticas con la finalidad de obtener buenas soluciones en tiempos razonables.

Una vez resuelto el problema de optimización surge otra problemática compleja lo cual tiene que ver con subjetividad ya que en la mayoría de las situaciones las soluciones proporcionadas tienen que ser analizadas y determinar de manera subjetiva cual o cuales de las alternativas de solución se apegan fielmente a las necesidades de la organización.

Esta actividad representa una gran carga cognitiva al Tomador de Decisiones (DM), el cual es el encargado de determinar cuál es el subconjunto de proyectos que serán financiados, por lo que incorporarle este conocimiento al algoritmo meta heurístico le ayudara a reducir la dimensionalidad del problema.

Esta incorporación de información se lleva a cabo mediante relaciones de preferencia las cuales son modeladas mediante relaciones de preferencia difusa y es representado mediante un modelo de programación lineal. Lo que se busca es analizar los algoritmos no solo en términos de buena calidad de solución sino que también presenten un buen compromiso.

Otro de los aspectos importantes en la investigación es hacer uso de la programación paralela para aprovechar mejor los recursos del equipo de cómputo lo que trae consigo una reducción de tiempo de ejecución.

1.2. Justificación.

La distribución de presupuesto en las empresas privadas como públicas, es de vital importancia para mejorar el funcionamiento de las mismas, por lo que una mala selección de proyectos puede afectar a la organización y esta no pueda alcanzar sus metas. Por este motivo se quiere resolver el problema de Cartera de Proyectos para mejorar los beneficios que se puedan obtener con un presupuesto definido.

Como las necesidades de las empresas no son iguales entre ellas no se puede proporcionar la misma solución a todas, aunque estas pertenezcan al mismo sector producido, ya que se rigen bajo diferentes criterios. Por esta razón se propone que al método de solución para el problema CPP se le incorporen preferencias para que las alternativas que se proporcionen se adapten a las necesidades específicas de una empresa en particular.

Aunque actualmente ya existen estrategias que aborden la solución de problemas de CPP que incorporen preferencias, éstas no se han abordado bajo el enfoque de programación paralela, todos hacen uso de programación secuencial para la solución de este problema.

Hoy en día, el trabajo experimental que permita evaluar el desempeño de nuevas estrategias de solución, basadas en preferencias, para CPP es extensivo y consume periodos largos de tiempo. Por lo tanto, el contar con un algoritmo que permita explotar infraestructura de hardware con varios núcleos de procesamiento podría ayudar a reducir el tiempo incluso hasta varios órdenes de magnitud. Por esta razón se considera un área de oportunidad el desarrollo de algoritmos paralelos que incorporen preferencias en la solución del problema CPP.

1.3. Objetivos

1.3.1. Objetivo General

Implementar algoritmo meta heurístico paralelo que incorpore preferencias de un tomador de decisión en la solución del problema de cartera de proyectos públicos con múltiples objetivos.

1.3.2. Objetivos Específicos.

En esta subsección se describen cada uno de los alcances que se persiguieron durante la

- Analizar métodos para evaluar preferencias de un tomador de decisión entre parejas de alternativas.
- Analizar algoritmos basados en preferencias que resuelvan el problema de CPP Multi-Objetivo.
- Revisión de la literatura científica para identificar estrategias que resuelven el problema de CPP de forma paralela.
- Análisis de patrones de diseño paralelos existentes aplicables en estrategias de solución para problemas Multi-Objetivo.
- Diseño de la propuesta bajo el paradigma paralelo para la solución del problema CPP.
- Implementación del algoritmo paralelo.
- Pruebas del algoritmo paralelo.
- Análisis del resultado.

1.4. Alcances y limitaciones.

Los alcances y limitaciones de este proyecto se mencionan a continuación:

- El problema de CPP incluirá solamente restricciones de presupuesto por área y región.
- La incorporación de preferencias será a través de uno de los métodos de outranking ELECTRE III y/o ELECTRE TRI.
- La solución del problema CPP será a través de un modelo subrogado de no más de 3 objetivos.

- El enfoque metaheurístico estará basado en una estrategia evolutiva.
- La implementación del diseño paralelo será en lenguaje C, haciendo uso de memoria compartida por medio de OpenMP, y de memoria distribuida por medio de MPI.

2. Marco Teórico.

En este capítulo se presentan los aspectos teóricos relacionados con el proceso de toma de decisiones y cómo influye el modelo de preferencias del DM en los problemas de optimización multi criterio en el caso particular de los problemas de CPP.

Además se describen las estrategias de paralelización que se han utilizado para resolver problemas de optimización y la métrica para medir el desempeño en comparación con un algoritmo secuencial.

2.1. Problema de Optimización.

Un problema de optimización es simplemente un problema para el cual existen varias soluciones y existe un mecanismo de comparación entre estas soluciones, de tal manera que existe un conjunto de soluciones candidatas que pueden ser comparadas entre sí. Desde el punto de vista matemático se puede definir como un problema de optimización P como una 3-tupla $P = (f, SS, F)$ de la siguiente manera (Duarte Muñoz & Pantrigo Fernandez, Juan Jose Gallego Carrillo, 2007):

$$P = \left\{ \begin{array}{l} \text{opt: } f(x) \\ \text{s. a.,} \\ x \in F \subset SS \end{array} \right\}$$

Donde $f(x)$ representa la función objetivo que se va optimizar (maximizar o minimizar) y esta se encuentra sujeto a un conjunto de restricciones las cuales son definidas por el problema a resolver, F es el conjunto de soluciones factibles y SS representa el espacio de soluciones.

Los problemas de optimización se clasifican en dos categorías según su codificación: aquellos donde la solución del problema esta codificada con números reales y por otro lado tenemos las soluciones que se encuentran codificadas con número enteros, los problemas que entran en esta clasificación se conocen como problemas de optimización combinatoria.

Estos problemas tienen como característica principal que existe un algoritmo exacto que permite que le permite encontrar la solución óptima para problemas mono objetivo y una o un conjunto de soluciones para problemas Multi Criterio (Duarte Muñoz & Pantrigo Fernandez, Juan Jose Gallego Carrillo, 2007).

2.2. Problemas de Optimización Multi Criterio.

Un problema de optimización consiste en minimizar o maximizar un conjunto de variables, buscando que la solución que represente los valores óptimos para cada una de las funciones objetivo. Además se deben cumplir ciertas restricciones dadas por el problema, en caso de existir, para evitar soluciones no válidas.

En (Coello, 2000) define un problema de optimización multi-objetivo de la siguiente manera:

$$\min x = \{f(x)\} \quad (\text{ec.1})$$

Sujeto a:

$$g_i \geq 0 \quad \forall i = 1, 2, 3, \dots, n \quad (\text{ec.2})$$

$$h_j = 0 \quad \forall j = 1, 2, 3, \dots, p \quad (\text{ec.3})$$

Donde f es un vector de k funciones objetivas $f = (f_1, \dots, f_k)$ y x es el vector solución $x = (x_1, \dots, x_r)$ de r variables, n es el número de restricciones de desigualdad; y p es el número de restricciones de igualdad.

La característica principal de estos problemas de optimización es que cuenta con la optimización simultánea con más de dos objetivos los cuales generalmente se encuentran en conflictos entre ellos (Coello, 2000; Fülöp, 2001).

En particular los algoritmos evolutivos Multi Objetivo (Multiple Objective Evolutionary Algorithms - MOEAs) permiten resolver problemas de optimización Multi Criterio los cuales construyen simultáneamente con un conjunto de posibles soluciones (población). Esto permite encontrar varios miembros del conjunto óptimo de Pareto en una sola ejecución del algoritmo, en lugar de tener que realizar una serie de pruebas separadas como en el caso de las técnicas tradicionales de programación matemática (Coello, 2008).

Aunque existen muchos enfoques de Optimización Evolutiva Multi Objetivo (Evolutionary Multi-Objective Optimization – EMOO), en la siguiente figura se presenta una clasificación con las técnicas más representativas divididas en dos generaciones (Coello, 2000) (Victoria Ruiz Martínez, 2016).

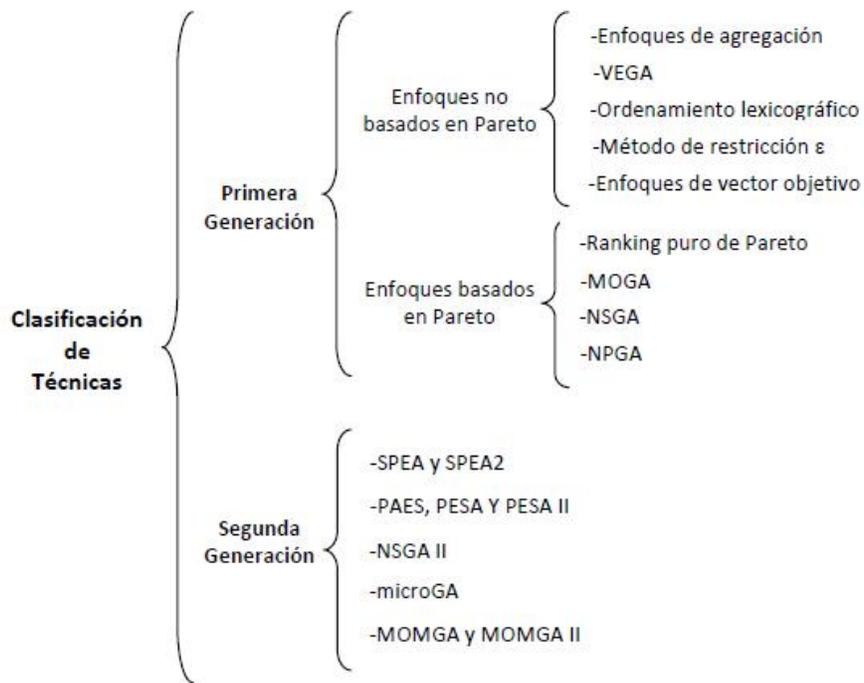


Figura 1.- Clasificación de técnicas EMOO(Victoria Ruiz Martínez, 2016).

Las técnicas EMOO pueden ser utilizadas para resolver el problema de Cartera de Proyectos, para este proyecto de investigación se hará uso de la técnica de primera generación llamada Ordenamiento Lexicográfico para medir la calidad de la solución en los algoritmo que incorporen preferencias.

Por otro lado tenemos que para los algoritmos que no incorporen preferencias se utilizara el algoritmo NSGA II, el problema a resolver es el CPP el cual pertenece a la familia de problema de Cartera de Proyectos se medirá la calidad de solución haciendo uso del enfoque basados en Pareto.

2.3. Cartera de Proyectos.

Es un problema de distribución de recursos el cual se encuentra presente en todos los aspectos de la sociedad y en las organizaciones de cualquier índole, en las cuales se presentan necesidades o problemáticas las cuales no pueden ser resueltas con el presupuesto disponible.

En las organizaciones las necesidades son subsanadas mediante la realización de proyectos que se encuentran en competencia por el recurso con la finalidad de ser financiados, siendo esta una actividad compleja(Place, Lattre, & Cedex, 2005).

El problema de cartera de Proyectos (CP) es una problemática de vital importancia para en el sector público y privado, Una cartera se encuentra definida por un conjunto de proyectos los cuales se

encuentran limitados a un presupuesto, donde cada uno de los proyectos seleccionados impactan en uno o más objetivos en diferente proporción.

Cuando se obtiene un conjunto de buenas soluciones y como existen factores de riesgo es cuando el DM tiene la responsabilidad de seleccionar la mejor cartera (Place et al., 2005), por lo que el DM tiene que hacer un análisis minucioso acerca de las necesidades específicas de la organización para determinar cuál o cuáles soluciones son en las que la organización se ve más beneficiada.

En este proyecto de investigación se resolverá el problema CP pero para la selección de proyectos públicos, donde la limitante de la cartera no solo depende del presupuesto a invertir si no el beneficio mínimo y máximo por categorías.

2.4. Cartera de Proyectos Públicos.

El problema de Cartera de Proyectos Públicos (CPP) es un caso particular del problema CP donde los proyectos son de carácter social, lo que quiere decir que se involucran aspectos que no pueden ser cuantificables en términos económicos, como por ejemplo bienestar social, seguridad y etc.

Este problema se puede considerar como una situación de toma de decisión en la cual el DM es el encargado de seleccionar un grupo de proyectos (cartera) que una institución llevará a cabo. El objetivo de este problema de decisión es elegir la “mejor” cartera satisfaciendo algunas restricciones presupuestales.

Formalizando estos conceptos, consideremos un conjunto de N proyectos, donde el i-ésimo proyecto es representado por un vector p-dimensional $F(x_i) = (f_1(x_i), f_2(x_i), f_3(x_i), \dots, f_p(x_i))$, donde cada $f_j(x_i)$, indica la contribución del proyecto i al j-ésimo objetivo. Cada objetivo indica el beneficio, el cual es, el número de personas que pertenecen a una categoría social (pobreza extrema, pobreza, clase media) y quienes reciben un nivel de beneficio (alto impacto, impacto medio, impacto bajo) del i-ésimo proyecto.

El i-ésimo proyecto corresponde a un área (salud, educación, etc.) señalada por a_i . Cada área tiene un límite presupuestal definido por el DM o por cualquier otra autoridad competente. Consideremos para cada área k, un límite inferior y superior, L_k y U_k respectivamente. Basado en esto, la restricción de cada área k de acuerdo a la Ecuación 4 es:

$$L_k \leq \sum_{i=1}^N x_i g_i(k) c_i \leq U_k \quad (\text{ec.4})$$

Donde $g(k)$ se define en la Ecuación 5 como:

$$g_i(k) = \begin{cases} 1 & \text{if } a_i = k, \\ 0 & \text{otherwise} \end{cases} \quad (\text{ec.5})$$

Además, cada proyecto corresponde a una región geográfica a la cual beneficiará. De la misma forma que las áreas, cada región tiene un límite inferior y superior como otra restricción que debe ser cumplida por una cartera factible.

La calidad de la cartera x es determinada por la unión de los beneficios de cada uno de los proyectos que la componen. Esto puede ser expresado en la Ecuación 6 como:

$$z(x) = (z_1(x), z_2(x), z_3(x), \dots, z_p(x)) \quad (\text{ec.6})$$

Donde $z_j(x)$ de la ecuación 6 en su forma más simple se define en la Ecuación 7 como:

$$z_j(x) = \sum_{i=1}^N x_i f_i(i) \quad (\text{ec.7})$$

Si expresamos por F_R la región de carteras factibles, el problema de cartera de proyectos es identificar una o más carteras que resuelvan:

$$\max_{x \in F_R} \{z(x)\} \quad (\text{ec.8})$$

Por otro lado, una cartera denominada x es un subconjunto de esos proyectos el cual es modelado generalmente como un vector binario $x = (x_1, x_2, \dots, x_n)$. En este vector, x_i es una variable binaria donde $x_i = 1$ si el i -ésimo proyecto es apoyado y $x_i = 0$ en otro caso.

Existe un presupuesto total, especificado en la ecuación 9, que la organización está dispuesta a invertir, el cual es denotado como B , y cada proyecto x_i tiene un costo asociado C_i , por lo que la sumatoria de los costos asociados a cada proyecto debe ser menor o igual al presupuesto total. Por lo tanto, las carteras están sujetas a la restricción presupuestal:

$$\left(\sum_{i=1}^N x_i * c_i\right) \leq B \quad (\text{ec.9})$$

El problema CPP es un caso especial de problemas de tipo *timetable* o programación de horarios; dichos problemas se han demostrado ser NP-Duros (Tim B. Cooper and Jeffrey H. Kingston, 1996), siendo esto un indicador de su complejidad, y justificante para incorporar algoritmos aproximados dentro de su repositorio de estrategias de solución.

Por otro lado tenemos que el problema CPP puede considerarse como como un problema de toma de decisiones debido a que si se decide si un proyecto es o no es financiado esto impacta de manera directa en el costo de la cartera, así como también en el impacto de los objetivos. Además se tiene que dado un conjunto de carteras surge otro problema de Toma de Decisiones y es evaluar cada una de las carteras con la finalidad de determinar cuál es la mejor cartera, en algunas situaciones influyen aspectos subjetivos para determinar cuál es la mejor cartera.

Estos problemas de optimización pueden ser resueltos por algoritmos exactos pero no es recomendable resolver los problemas porque su tiempo de ejecución incrementa de manera exponencial, por lo que el uso de meta-heurísticas es más recomendable porque provee buenas soluciones en tiempos razonables.

2.5. Meta-Heurísticas.

Estos algoritmos tienen como característica principal que con el paso del tiempo de su ejecución de estos algoritmos empiezan a mejorar la calidad de cada una de las soluciones generadas, lo que implica que las soluciones se van acercando a la o a las soluciones óptimas del problema.

El término meta-heurísticas es la composición de dos palabras que provienen del vocablo griego el término griego heurística proviene del vocablo *heuriskein* que podría traducirse como encontrar descubrir o hallar (Duarte Muñoz & Pantrigo Fernández, Juan José Gallego Carrillo, 2007). Podemos definir a una heurística como:

“Procedimientos simples, a menudo basados en el sentido común, que se supone que obtendrán una buena solución (no necesariamente óptima) a problemas difíciles de un modo sencillo y rápido” (Zanakis & Evans, 1981).

Por otro lado tenemos que el término “meta” es un prefijo (en inglés) que se podría traducir como “más allá de”. Una definición de meta-heurística es:

“Las meta-heurísticas son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria en los que los heurísticos clásicos no son efectivos. Las meta-heurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos” (Osman & Kelly, 1997).

Una de las consideraciones que se debe de tener para estudiar y analizar las meta-heurísticas es que está debe contener conocimiento acerca de la función objetivo, lo cual le sirve a la estrategia de solución como mecanismo de comparación entre soluciones y así determinar cuál o cuáles son las mejores soluciones encontradas.

Para los algoritmos tradicionales (La calidad de solución depende solo de los beneficios obtenidos en los objetivos) la técnica utilizada es la dominancia.

2.6. Algoritmos Genéticos.

Estos algoritmos entran en la clasificación de algoritmos evolutivos (EA, por sus siglas en inglés Evolutionary Algorithms) los cuales ayudan a resolver problemas de optimización combinatoria, estos EA se basa en la idea neo-darwiniana de la evolución de las especies la cual se puede resumir de la siguiente manera:

“Los individuos que tienen una mejor adaptación al medio tienen una probabilidad más elevada de vivir más tiempo, con lo que tendrán más posibilidades de generar descendencia que herede sus buenas características. En cambio, los individuos con peor adaptación al medio, tienen menos probabilidad de sobrevivir, por lo que tendrán menos oportunidades de generar descendencia y probablemente acaben

extinguiéndose”(Duarte Muñoz & Pantrigo Fernandez, Juan Jose Gallego Carrillo, 2007).

Los elementos que interviene en el proceso de solución del problema mediante el uso de un Algoritmo Genético (GA por sus siglas en inglés Genetic Algorithm) son los siguientes(Duarte Muñoz & Pantrigo Fernandez, Juan Jose Gallego Carrillo, 2007):

- **Generación de una población inicial.** Este proceso tiene como finalidad crear un conjunto de soluciones iniciales, las cuales mediante el proceso ayudaran a mejorar la calidad de la solución. Generalmente esta población es generada aleatoriamente.
- **Representación de la solución.** Describe las características de una solución lo cual ayudara a comparar soluciones mediante el beneficio de la solución (fenotipo) y la codificación de la solución (genotipo) normalmente el genotipo pueden ser cadenas, arreglos binarios o números reales.
- **Función de Evaluación.** Ayuda a determinar la calidad de la solución de los individuos en la población.
- **Operadores Genéticos.** Son métodos probabilísticos que ayudan a construir nuevas soluciones (individuos) estas suelen ser independientes de la representación, los operadores genéticos comúnmente utilizados son:
 - **Cruza:** esta técnica en la sustitución de un conjunto de genes de los padres los cuales son asignados a la nueva solución (hijo).
 - **Mutación:** esta técnica probabilística tiene como finalidad hacer cambios esporádicos en la solución hija, lo cual ayuda a mantener la diversidad entre las soluciones y a explorar nuevas zonas en el espacio de búsqueda.
- **Selección.** Es un mecanismo que nos ayuda a identificar cuáles fueron las mejores soluciones que existen en la población y que están sean preservadas para futuras iteraciones.

2.7. Algoritmos de Optimización por Colonia de Hormigas

La teoría de optimización por colonia de hormigas (Ant Colony Optimization, aco), fue introducida por Marco Dorigo en los inicios de 1990 como herramienta para la solución de problemas de optimización complejos (Robles Algarín, 2010).

Esta meta heurística está inspirada en la observación del comportamiento de hormigas reales (Alejandro Arito, 2010; Rivera Zárate, Gómez Santillan, & René, 2015), la cual hace uso de técnica probabilística para solucionar problemas computacionales los cuales no pueden reducirse solo a buscar los mejores caminos o rutas en grafos.

Por otro lado tenemos que las hormigas son insectos sociales, ya que hacen uso de la inteligencia colectiva para evitar que la colonia se extinga, por lo que este comportamiento está dirigido más hacia la supervivencia de la colonia como un todo que al de una simple componente individual de la colonia.

Diferentes estudios han demostrado que la comunicación de las hormigas a través de caminos con feromonas les permite encontrar las rutas más cortas entre su nido y las fuentes de alimentos (Alejandro Arito, 2010).

Los algoritmos ACO han sido formalizados como una meta heurística de optimización de combinatoria por lo que, han sido usados para resolver muchos de estos problemas.

Esta meta heurística está conformada por un conjunto de agentes computacionales (hormigas) que trabajan de manera conjunta para poder comunicarse a través de rastros de feromonas artificiales (Robles Algarín, 2010). Los componentes de un algoritmo de hormigas se presentan a continuación:

- **Inicialización de la Colonia.** Este proceso tiene como finalidad inicializar la tabla de feromonas asignándole un peso a cada uno de los arcos del grafo, esta matriz de feromonas normalmente es inicializada en 1 para que todos los nodos tengan la misma probabilidad de ser aceptados en la primera iteración.
- **Construcción de soluciones.** Este proceso tiene como finalidad construir una nueva solución a partir de la información de la tabla de feromonas haciendo uso de la regla de decisión proporcional.
- **Búsqueda Local.** Este mecanismo es opcional en el algoritmo propuesto por Dorigo, pero tiene como finalidad encontrar una mejor solución a partir de alteraciones que se le hacen a la solución.
- **Actualización de Feromonas.** Mecanismo tiene como finalidad actualizar la información generada por la colonia, los cambios que son utilizados para la actualización de feromonas son las siguientes:
 - **Intensificación de feromonas.** Este proceso tiene como finalidad incrementar el rastro de feromonas de los arcos que se encuentran en una buena solución, lo que quiere decir para que una hormiga pueda intensificar la tabla de feromonas ésta debió de haber encontrado una solución con una buena calidad.
 - **Evaporación de Feromonas.** Este proceso tiene como finalidad eliminar gradual mente los rastros de feromonas de los nodos del grafo que dejan de ser visitados por las hormigas, con la finalidad de darle elitismo
 - al algoritmo.

Tanto como GA y ACO hacen uso del cálculo de la función de evaluación o función de aptitud para ayudar a que el algoritmo evolucione mejorando así la calidad de las soluciones que son generadas

conforme van transcurriendo las iteraciones, he aquí la importancia de la selección de la función de evaluación principal mente para los algoritmos Multi objetivo.

2.7.1. Función de Aptitud para Problemas Multi Objetivo

Es el mecanismo que le asigna una ponderación a una solución en específico, a esta ponderación se le denomina calidad de solución. La cual ayuda a determinar cuál o cuáles son las mejores soluciones encontradas.

En el caso particular de los problemas multi objetivo se hacen uso de indicadores de desempeño para determinar qué tan buena es la calidad de una solución respecto, estos indicadores pueden ser los siguientes:

- Dominancia.
- Estimación de Densidad.
- Métodos de sobre clasificación difusa.

En la literatura científica el indicador más reportado para la solución de problemas multi objetivo son para problemas que no incorporan preferencias es el de la Dominancia el cual hace uso de los beneficios obtenidos en cada uno de los objetivos involucrados en el problema.

2.7.2. Dominancia.

Es una comparación binaria que ayuda a determinar si una alternativa es mejor que otra en términos de beneficio obtenido en los objetivos, esta comparación es realizada objetivo por objetivo. Considerando las soluciones a, b se puede concluir lo siguiente:

- a domina a b , si y solo en al menos un objetivo i de la solución a mejora al objetivo i de la solución b y en el resto de los objetivo es al menos igual.
- a no domina a b , si y solo si existe al menos un objetivo i en la solución b que mejora ala solución a .

Esta comparación es utilizada en los algoritmos que hacen uso del método de Non Fast Dominated con la finalidad de agrupar en conjuntos las soluciones que han sido encontradas por el algoritmo evolutivo. Por otro lado tenemos el indicador de estimación de la densidad el cual es utilizado junto al algoritmo Non Fast Dominated para el algoritmo NSGA II.

2.7.3. Fast Non-Dominated Sorting.

Con el fin de ordenar la población de tamaño N de acuerdo al nivel de no dominancia, cada solución debe ser comparada con toda solución en la población para encontrar si es dominada. Esto requiere $O(mN)$ comparaciones para cada solución, donde m es el número de objetivos(Victoria Ruiz Martínez, 2016).

Cuando el proceso continúa para encontrar los miembros del primer frente de clase no dominada por todos los miembros, la complejidad total es $O(mN^2)$. En esta

etapa, todos los individuos en el primer frente son encontrados (Victoria Ruiz Martínez, 2016).

Con la finalidad de encontrar a los individuos del siguiente frente, las soluciones del primer frente son temporalmente descontadas y el proceso se lleva a cabo nuevamente. En el peor caso, la tarea de encontrar el segundo frente también requiere $O(mn^2)$ (Victoria Ruiz Martínez, 2016). En la siguiente figura se describe el algoritmo *fast non dominated sorting*.

```

Fast-nondominated-sort (P)
1   para cada  $p \in P$  hacer
2   para cada  $q \in P$  hacer
3   si  $(p \prec q)$  entonces
4    $S_p = S_p \cup \{q\}$ 
5   en caso contrario - si  $(q \prec p)$ 
   then
6    $n_p = n_p + 1$ 
7   si  $n_p = 0$  entonces
8    $F_1 = F_1 \cup \{p\}$ 
9    $i = 1$ 
10  mientras  $F_i \neq 0$  hacer
11   $H = 0$ 
12  para cada  $p \in F_i$  hacer
13  para cada  $q \in S_p$  hacer
14   $n_q = n_q - 1$ 
15  if  $n_q = 0$  entonces  $H = H \cup \{q\}$ 
16   $i = i + 1$ 
17   $F_i = H$ 

```

Figura 2.- Algoritmo *fast non dominated sorting*

2.7.4. Estimación de la Densidad.

Este indicador determina la densidad que existe entre un conjunto de las soluciones alrededor de un punto en particular de este conjunto, el proceso consiste en calcular la distancia promedio

existente entre 2 puntos en cada lado del mismo, considerando cada uno de los objetivos como componentes lineales.

El valor sirve como un estimado del tamaño del cuboide más grande el cual encierra cada uno de los puntos sin incluir otro punto(solución) A esto se le denomina la distancia de a esto se le llama distancia Crowding)(Victoria Ruiz Martínez, 2016). Éste algoritmo se describe en la Figura 3.

```
crowding-distance-assignment (I)
1   l = |I|
2   para cada i, hacer set l [i]distance =0
3   para cada objective m hacer
4     l=sort (l, m)
5     for i =2 to(l-1)
6     l [i]distance = l [i]distance + (l [i +1]).m – l [i-1].m)
```

Figura 3.- Distancia de Crowding.

De la misma manera que el indicador de la dominancia este indicador tiene como finalidad hacer la evaluación entre las soluciones tomando solo en consideración únicamente los objetivos involucrados en el problema.

Por otro lado tenemos que uno de los indicadores más reportados en la literatura que hacen uso de la articulación de preferencias, son los métodos de sobre clasificación haciendo uso de información generada durante proceso de Toma de decisiones con la finalidad de simular a un Decisor a través de su modelo preferencial.

2.8. Toma de Decisiones.

Es un proceso de aprendizaje natural o estructurado mediante el cual se tiene que elegir entre dos o más alternativas de solución que ayuden a resolver una problemática específica en la que se le presente a la(s) persona(s), el cual es conocido comúnmente como Tomador de Decisiones(Rivera Zárate et al., 2015).

Tenemos que para poder tomar una buena decisión es necesario tener la información necesaria con la finalidad de poder visualizar los posibles escenarios y contemplar así el impacto positivo, el riesgo y las consecuencias que trae con sigo esta decisión(Fülöp, 2001).

Para tomar una decisión, no importa su naturaleza, es necesario conocer, comprender, analizar un problema, para así poder darle solución; en algunos casos por ser tan simples y cotidianos, este proceso se realiza de forma implícita y se soluciona muy rápidamente, pero existen otros casos en

los cuales es necesario realizar un proceso más estructurado que puede dar más seguridad e información para resolver el problema(Sapag, 2000; URBINA & CASTELLANOS, 2006).

Para hacer el análisis y la evaluación de las carteras generadas se realiza de manera subjetiva es decir que se pueden emitir opiniones, juicios, prioridades, etc. del DM, los cuales representan las necesidades específicas de una organización lo cual se conocen como preferencias.

Por lo que podemos definir como proceso de toma de decisiones a la situación en la cual el Tomador de Decisiones tiene que decidir cuál o cuáles son las carteras se apegan fielmente a sus preferencias. Lo que implica que tomar una decisión se necesita tener al menos dos soluciones para poder discernir cuál es la mejor alternativa. (Fülöp, 2001).

2.8.1. Modelos de Preferencia.

Son modelos que ayudan a determinar cuál es la cartera que se adapte a las, metas, objetivos, deseos del DM tanto como sea posible (Fülöp, 2001; Place et al., 2005). Por esta razón el DM es el actor central que existe en todo problema de decisión, ya que el modelo preferencial expresa las necesidades.

El DM cuenta con un sistema de preferencias, el cual representa con valores las prioridades, metas, objetivos, deseos, etc. Las cuales impactan de manera directa en la selección de atributos, ponderándolos de manera subjetiva esto tiene impacto en el proceso de solución(Rodríguez, Reyes, & López, 2010).

Este modelo es usado mediante relaciones binarias de sobre clasificación (Outranking) las cuales se denotaran con S , lo que indica que “Una solución es al menos tan buena como otra”. Siendo a y b dos alternativas se tienen que considerar los siguientes escenarios(Mousseau, Slowinski, & Zielniewicz, 2000; Place et al., 2005):

- $aSb \wedge \neg bSa$, se puede decir que aPb (a se prefiere estrictamente sobre b).
- $bSa \wedge \neg aSb$, se poder decir que bPa (b se prefiere estrictamente sobre a).
- $aSb \wedge bSa$, podemos decir que aIb (a es indiferente con b).
- $\neg aSb \wedge \neg bSa$ podemos decir que aRb (a, b no se pueden comparar).

Estas relaciones son las relaciones son necesarias para clasificar, jerarquizar o seleccionar las alternativas basadas en las preferencias del DM, Estas problemáticas que pueden ser solucionadas por los métodos ELECTRE.

2.8.2. Métodos ELECTRE.

Los métodos ELECTRE ayudan a construir una o varias relaciones difusas de sobre clasificación las cuales ayudan a analizar las alternativas de solución basadas en las preferencias de un DM, la construcción de estas relaciones se basan principalmente en dos conceptos(Mousseau et al., 2000; Pena, Rebollo, Oliveras, & Mateu, 2007):

- **Concordancia.** Quiere decir que en la mayoría de los criterios hace cierta la aseveración aSb , el valor es una relación de pertenencia.
- **No Discordancia.** Representa el grado de desacuerdo que se opone fuertemente a la aseveración aSb .

Para poder determinar el nivel de concordancia y discordancia es necesario identificar la el impacto de cada uno de los criterios, para los métodos ELECTRE se hace uso de los umbrales de Peso y Veto. El peso no depende de alguna escala en particular, en muchas ocasiones es necesario normalizar este umbral para poder ser utilizados por el método ELECTRE(Zak & Kruszyński, 2015).

Por otro lado el umbral Veto representa la importancia de un objetivo que contradice la aseveración “ a es al menos tan buena como b ” esto ocurre cuando la comparación de las alternativas a, b superan este umbral en alguno de los objetivos(Place et al., 2005).

Los umbrales de discriminación son los elementos que ayudan a la estructura de preferencias del DM ya que estos determinan el nivel de aceptación por cada uno de los objetivos, lo que ayuda a determinar la relación binaria entre dos alternativas, estos umbrales se describen a continuación:

- p es un vector que representa el umbral de preferencia. Este umbral representa el valor numérico por el cual el DM prefiere una alternativa sobre otra.
- q es un vector que representa el umbral de indiferencia. Es el valor numérico por el cual el DM es indiferente ante dos alternativas.

Los problemas que se pueden resolver mediante los métodos Electre son Selección, Jerarquización, Clasificación (Figueira, Greco, & Ehrogott, 2005) donde el principal mecanismo de comparación son las preferencias del DM.

La selección es la problemática con la que se enfrenta el DM cuando requiere determinar la mejor solución basado en su criterio, esta problemática tiene como objetivo ayudar al DM a generar un subconjunto de alternativas lo más pequeño posible, y en relación con a este sub-conjunto se pueda seleccionar la mejor alternativa.

Como resultado de este proceso se obtiene cuáles son buenas alternativas y cuales son malas alternativas de solución, los Métodos ELECTRE que ayudan con esta problemática son(Figueira et al., 2005; Place et al., 2005):

- ELECTRE I.
- ELECTRE IV.
- ELECTRE Is.

La tarea de jerarquización tiene como objetivo ordenar todas las alternativas de solución de la mejor a la peor basadas en su modelo preferencial. Los métodos ELECTRE que apoyan con esta problemática son(Figueira et al., 2005; Place et al., 2005):

- ELECTRE II.
- ELECTRE III.
- ELECTRE IV.

Para solucionar la problemática de clasificación es necesario definir cada una de las categorías se encuentra definida por medio de un conjunto de perfiles de referencia, y cada una de las categorías queda definida por medio de las comparaciones entre los perfiles de referencia estableciendo límites (superior, inferior)(Figueira et al., 2005).Estos límites ayudan a definir clases o categorías a las cuales se les asignan variables lingüísticas.

Los resultados de esta problemática se expresan utilizando la convicción de Asignado o no Asignado a una clase en particular, el método ELECTRE que ayuda con esta problemática ELECTRE TRI(Figueira et al., 2005).

Para mejorar el proceso de Selección, Jerarquización y clasificación es necesario determinar tener una buena definición de los umbrales de discriminación, que concuerde con las preferencias del DM(Figueira et al., 2005; Place et al., 2005).

El proceso de estimación de Parámetros se obtiene a través de la interacción entre el DM y los analistas, en donde el DM expresa la información de sus Preferencias de cada uno de los criterios de una solución. Las técnicas de estimación de parámetros son utilizadas para determinar los umbrales de discriminación las técnicas más reportadas en la literatura son las técnicas directas y las técnicas indirectas(Figueira et al., 2005):

Las técnicas directas tienen como característica principal que el DM proporciona de manera directa los parámetros de preferencia por esta razón la mayoría de las implementaciones de los métodos ELECTRE no hacen uso de esta técnica para definir los parámetros de preferencia(Figueira et al., 2005).

Por otro lado las técnicas de Estimación indirecta no requieren de DM's para proporcionar respuestas a preguntas relacionadas con los valores de los parámetros de preferencia, los valores de los parámetros se asignan indirectamente dando respuesta a unas preguntas relacionadas con los parámetros de preferencias (Figueira et al., 2005).

En la actualidad se han desarrollado técnicas de Estimación de parámetros para los métodos (Figueira et al., 2005). Para hacer el cálculo del grado de credibilidad se hizo uso de la metodología del método ELECTRE TRI. En el proyecto de investigación el cálculo del grado de credibilidad se utiliza para los algoritmos que incorporan preferencias.

2.8.3. Modelo de Sobre Clasificación Difusa.

Es un modelo de programación lineal multi objetivo el cual es obtenido a partir de las relaciones de preferencia generadas, donde cada uno de los objetivos de este modelo es representado por una relación o la combinación de relaciones de preferencia. Las relaciones propuestas por Fernández (2013) son las siguientes:

- Preferencia Estricta.
- Indiferencia.
- Preferencia Débil.
- K-Preferencia.
- Comparabilidad.
- No Preferencia.

La preferencia estricta se presenta cuando existe evidencia suficiente para que el DM pueda justificar que una alternativa a es mejor que la alternativa b , en términos de del grado de credibilidad $\sigma(a, b)$, se dice que una alternativa de solución cumple al menos una de las siguientes condiciones(Fernández, Lopez, Mazcorro, Olmedo, & Coello Coello, 2013; Rivera Zárata, Gómez Santillan, & René, 2015; Rodríguez et al., 2010):

- $(x \text{ domina } a \text{ y } y)$
- $\sigma(a, b) \geq \lambda \wedge \sigma(b, a) < 0.5$
- $\sigma(a, b) - \sigma(b, a) \geq \beta$

La indiferencia se denota aIb la cual representa un alto grado de equivalencia y homogeneidad, por lo que el DM no puede decidirse por una de ellas(Fernández et al., 2013; Rivera Zárata et al., 2015). En términos de $\sigma(a, b)$ esta relación se encuentra definida de la siguiente manera:

- $\sigma(a, b) \geq \lambda \wedge \sigma(b, a) \geq \lambda$
- $|\sigma(a, b) - \sigma(b, a)| \leq \epsilon$

La preferencia débil Se representa como aQb , la cual modela un estado de duda entre la preferencia estricta y la indiferencia, la cual se encuentra definida por medio de las siguientes conjunciones(Fernández et al., 2013; Rivera Zárata et al., 2015; Rodríguez et al., 2010):

- $\sigma(a, b) \geq \lambda \wedge \sigma(a, b) \geq \sigma(b, a)$
- $\neg aPb \wedge \neg bIa$

La incomparabilidad se representa como aRb , se dice que existe tal grado de heterogeneidad entre las soluciones que no pueden ser comparadas. Se dice que las soluciones son incomparables entre sí cuando(Fernández et al., 2013; González & Castillo, 2001; Rodríguez et al., 2010):

- $\sigma(a, b) < 0.5 \wedge \sigma(b, a) < 0.5$

La K-preferencia se denota como aKb esta representa un estado de duda entre la preferencia estricta y la incomparabilidad, esta se encuentra definida por medio de la siguiente clausula(Fernández et al., 2013; Rivera Zárata et al., 2015; Rodríguez et al., 2010).

$$\triangleright (0.5 \leq \sigma(a, b) < \lambda) \wedge (\sigma(b, a) < 0.5) \wedge (\sigma(a, b) - \sigma(b, a) > \frac{\beta}{2})$$

La No preferencia representa que no existe ninguna relación de preferencias entre dos alternativas de solución, y se representa por $a \sim b$ y esta definida por las siguientes clausulas(Fernández et al., 2013; Rivera Zárata et al., 2015; Rodríguez et al., 2010):

$$\triangleright \neg aPb \wedge \neg bPa$$

$$\triangleright \neg aQb \wedge \neg bQa$$

$$\triangleright \neg aKb \wedge \neg bKa$$

$$\triangleright \neg aIb$$

$$\triangleright \neg aRb$$

El modelo utilizado en este proyecto de investigación para los algoritmos que incorporan las preferencias de un decisor el modelo propuesto por Fernández et. al(2013), el cual genera un modelo sub-rogado de tres objetivos los cuales se obtienen a partir de la combinación de reglas de preferencia. Este modelo ayuda caracterizar la región de interés del DM en la cual existe una mayor probabilidad de identificar “La Mejor Cartera”.

2.9. Estrategias de Paralelización para Meta heurísticas

Actualmente existe en la comunidad científica la necesidad de resolver problemas que requieren un gran costo computacional, como por ejemplo los problemas de optimización combinatoria con algoritmos meta heurísticos.

Actualmente las técnicas de paralelización de algoritmos las cuales definen un conjunto de metodologías sumamente utilizadas para la resolución de problemas de gran escala y alta complejidad computacional. Entre estos problemas se cuentan los NP-Hard de optimización combinatoria, que debido a la imposibilidad de encontrar las soluciones óptimas son resueltas con heurísticas en búsqueda de una buena solución aproximada De la fuente, (2005), divide las estrategias de paralelización de la siguiente manera:

- Maestro - esclavo.
- Grano grueso.
- Grano fino.
- Grano grueso y fino.
- Grano grueso y maestro - esclavo.
- Grano grueso a dos niveles.

2.9.1. Maestro – Esclavo (paralelización global)

El esquema más simple de paralelización es la paralelización global. Un procesador central es el encargado de la selección, mientras que los procesadores esclavos son los encargados de hacer mutación y/o evaluación de la aptitud como se muestra en la Figura 4. Aparentemente se ve como un algoritmo secuencial, sin embargo, su tiempo de ejecución es mucho menor (De la fuente, 2005).

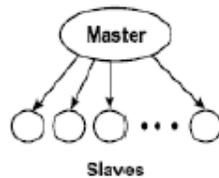


Figura 4.-Estrategia Maestro Esclavo.

2.9.2. Grano Gueso

En este esquema se divide la población en islas, las cuales se ejecutaran en paralelo, como se muestra en la Figura 5. Entre estas islas se hacen intercambios para introducir cierta diversidad, evitando que las soluciones queden en óptimos locales (De la fuente, 2005).

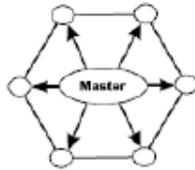


Figura 5.-Estrategia de Grano Gueso

2.9.3. Grano Fino.

Esta estrategia se limita a componerse de 1 o 2 individuos, los cuales solo pueden interactuar con sus vecinos cercanos así como se ilustra en la Figura 6, es decir, se adopta el concepto de vecindad. Este esquema es dirigido a equipos con un alto paralelismo (De la fuente, 2005).

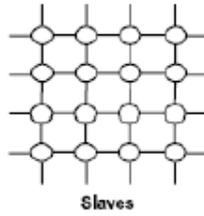


Figura 6.-Estrategia de Grano Fino.

2.9.4. Grano Grueso y Fino

Es una estrategia híbrida, en donde se combina el concepto de vecindad y el de población en islas, ayudando a salir de los óptimos locales, esta estrategia se puede visualizar gráficamente en la Figura 7(De la fuente, 2005).

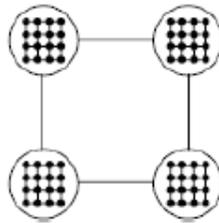


Figura 7.- Estrategia de Grano Grueso y Fino.

2.9.5. Grano Grueso y Maestro – Esclavo.

También, es una estrategia híbrida la cual se compone de la estrategia Maestro esclavo tomando en cuenta la idea de la estrategia de grano grueso, igualmente, para favorecer a los óptimos globales como se muestra en la Figura 7(De la fuente, 2005).

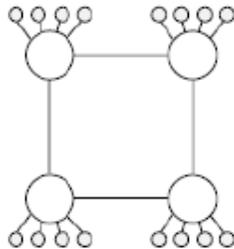


Figura 8.-Estrategia de Maestro-Esclavo y Grano Grueso

2.9.6. Grano Grueso a Dos Niveles.

En esta estrategia se utiliza doblemente el criterio de grano grueso, formando islas dentro de una isla, haciendo más variadas las soluciones obtenidas, la cual se puede ver gráficamente en la Figura 9.

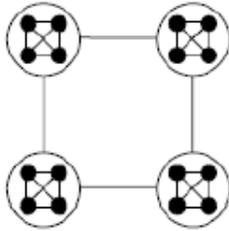


Figura 9.- Estrategia de Grano Grueso a Dos Niveles.

2.9.7. Paralelismo por Tareas.

Este patrón consiste en descomponer el problema en una colección de tareas que pueden ser ejecutadas de manera concurrente, como se puede mejorar esta concurrencia de manera eficiente.

Para la aplicación de este patrón es necesario identificar tareas que puedan ser ejecutadas de manera concurrente. (Timothy G. , Beverly A. , & Berna L., 2004)

2.9.8. Divide y vencerás.

Este problema consiste en dividir un problema secuencial con la estrategia de divide y vencerás de manera eficiente. Con esta estrategia se puede solucionar un problema mediante la división de este en problemas más pequeños, los cuales son solucionados de manera independiente para posteriormente unir cada una de las soluciones de los sub-problemas (Timothy G. , Beverly A. , & Berna L., 2004).

2.10. Aceleración o Speedup, estrategia para medir el desempeño de un Algoritmo Paralelo.

El beneficio en cuestión de rendimiento de una aplicación utilizando alguna librería para implementar programas paralelos, como OpenMP o MPI, depende totalmente de la medida en que

pueda ser paralelizada. La ley de Amdahl especifica la velocidad máxima que se puede esperar paralelizando porciones de un programa (Sanjay Kumar Sharma, 2012).

Sea (S) la velocidad máxima de un programa está dado por la siguiente ecuación:

$$S = \frac{1}{(1-f)} + \left(\frac{f}{n}\right)$$

Donde f es la fracción del tiempo de ejecución tomando la porción del código que puede ser paralelizada y N es el número de procesadores sobre el cual se ejecuta la porción paralela del código.

La métrica utilizada para la evaluación del rendimiento del algoritmo paralelo es la aceleración S_p , la cual se define de la siguiente manera:

$$S_p = T_1/T_p$$

Donde T_1 denota el tiempo de ejecución del algoritmo secuencial en la máquina de un solo procesador, y T_p es el tiempo de ejecución del algoritmo paralelo en la máquina del procesador P.

En otras palabras, el acelerar se refiere a cuanto el algoritmo paralelo es más rápido que el algoritmo secuencial. La aceleración lineal o ideal se obtiene cuando $SP=P$. En casos particulares es posible encontrar extensiones o modificaciones a la aceleración para medir el desempeño de algoritmos paralelos. (Sanjay Kumar Sharma, 2012)(Kathavate & Srinath, 2014).

3. Trabajos Relacionados.

Debido a la gran aplicabilidad del problema de CP se hizo una revisión en la literatura científica con la finalidad de ubicar, posicionar y dimensionar el impacto del proyecto de investigación de esta tesis. Dentro de las propuestas analizadas para la solución del problema de CP se hizo especial énfasis a los problemas enfocados a los proyectos públicos.

Por otro lado tenemos que una de las áreas de investigación que se abarcan en este proyecto es en el área de toma de decisiones ya que se busca simular el proceso de comparación entre alternativas de solución mediante un ILP (Modelo de Programación Lineal), y así identificar las alternativas que persigan las preferencias del DM.

Finalmente tenemos que otra de las aportaciones es el análisis del desempeño de los algoritmos de solución que hacen uso de la programación paralela, en la revisión de la literatura científica se hizo énfasis en la arquitectura de memoria compartida y memoria distribuida.

3.1. Manejo de Preferencias.

Para iniciar con el análisis de los trabajos relacionados se describen a manera general las aportaciones realizadas en el área de toma de decisiones, haciendo énfasis en los modelos de sobre clasificación.

En (Fülöp, 2001) se hace una implementación del algoritmo Electre Tri al cual el DM le hace una retroalimentación a la aplicación para definir el modelo de preferencias y perfiles de referencia, con finalidad de clasificar un conjunto de alternativas de solución.

En (Mousseau et al., 2000) se presenta una implementación del Algoritmo ELECTRE TRI en cual integra funciones que le permiten al DM interactuar en el proceso de estimación de parámetros preferencias. Por medio de inferencias en los parámetros de preferencias del ELECTRE TRI a partir de ejemplos de asignación.

En (Zak & Kruszyński, 2015) hacen uso de los métodos Analytic Hierarchy Process (AHP) y los métodos ELECTRE III/IV aplicado a 18 proyectos de transporte urbano resolviéndolos con estos métodos para generar la jerarquización e hicieron una comparación entre los resultados generados con los métodos utilizados.

En (Eduardo & Carvalho, 2011) se hace un análisis de las micro y pequeñas empresas para determinar su flexibilidad según con las capacidades y la adaptación a el sector comercial, teniendo como finalidad clasificar el desempeño de 4 grupos de comerciantes haciendo uso del método ELECTRE III, basados en los siguientes criterios:

- G1: Agrupamiento de productores en telares.
- G2: Agrupamiento de productores de estaño.
- G3: Agrupamiento de productores artículos de hierro.
- G4: Agrupamiento de productores de artículos de madera

Donde cada Agrupamiento fue considerado como una alternativa, formando así un grupo de alternativas.

En (Tam, Tong, & Lau, 2003) se hace uso del método ELECTRE III para facilitar al DM de construcción, para la selección del tipo construcción que se requiere para construir la planta de construcción en el caso particular de la aplicación de vibradores de concreto.

En (Figueira et al., 2005) se en listan algunos sectores en los cuales se han aplicado las técnicas y la aplicación de los métodos Electre, los cuales se muestran a continuación:

- Agricultura y Manejo Forestal.
- Energía.
- Medio Ambiente y Gestión del Agua.
- Financiar.
- Militar.
- Selección de proyectos (licitación).
- Transporte.

Basado en la revisión de la literatura científica se puede concluir que los métodos más utilizados para los análisis de preferencias son los de la familia de los métodos Electre de manera más puntual el método Electre III y Electre Tri.

3.2. Algoritmos paralelos.

En esta subsección se hace una revisión en la literatura científica para identificar áreas de aplicación de algoritmos paralelos, no solo aplicados en el área de Optimización sino también aplicaciones de características de aspecto general.

En (Nesmachnow, 2004) hacen uso de algoritmos paralelos evolutivos para resolver la problemática del diseño de redes de comunicaciones de alta conectividad topológica, abordando el problema de Steiner Generalizado, el cual consiste en dada un red de comunicaciones propone en diseñar una sub-red con mínimo costo.

En (Rafael Arturo Trujillo Rasúa, Antonio M. Vidal Maciá, 2009) se abordan problemas de optimización discreta con la finalidad de resolver el problema de la decodificación de señales en sistemas de comunicaciones inalámbricas haciendo uso de las siguientes arquitecturas paralelas:

- Memoria Compartida.
- Memoria Distribuida.
- Esquemas Híbridos.

Haciendo uso de las librerías de OpenMP, MPI para la implementación del algoritmo de solución.

En (Genis-triana & Rivera-lópez, 2010) hacen uso de dos algoritmos paralelos para la codificación de video basado en el estándar H.264/AVC con balanceo predictivo, con este estándar se obtiene una mejor calidad en la imagen que se visualizan en los videos, haciendo uso de la librería MPI.

En (Cortés, n.d.) hacen uso de un algoritmo genético paralelo con la finalidad de hacer una predicción precisa, para la minimización de daños ocasionados por los incendios forestales, implementando un algoritmo híbrido entre OpenMP y MPI con la finalidad de disminuir el tiempo de predicción.

En (Santander-jim, Vega-rodr, Juan, & Juan, n.d.) hacen uso de un sistema paralelo híbrido bajo el esquema maestro-esclavo basado en las tecnologías MPI y OpenMP, para darle solución al problema de la inferencia filogenética, este problema pertenece a la disciplina de la Bioinformática. En este trabajo se implementó el algoritmo NSGA II.

En (Ni et al., 2015) se hace una simulación sísmica haciendo uso de una arquitectura híbrida entre CPU y GPU, haciendo uso de CUDA para la ejecución de esta arquitectura.

En (Fern & Toro, n.d.) este trabajo se intenta mejorar el desempeño del problema multi-objetivo por medio de un algoritmo evolutivo para resolver una variante el problema de Enrutamiento de Vehículos con ventajas temporales.

Según con los artículos analizados se puede concluir que la programación paralela mejora el tiempo de ejecución para resolver los problemas de optimización, y la aplicación de algoritmos híbridos y la paralelización de los algoritmos secuenciales el éxito depende de la naturaleza del problema.

3.3. Algoritmos que incorporen preferencias para la solución del problema CPP.

Finalmente también se revisó en la literatura proyectos que hagan uso de preferencias con la finalidad de identificar cual es la estrategia más utilizada para la incorporación de preferencias para la solución del problema CPP.

En (Rodríguez et al., 2010) se hace un análisis del algoritmo NOSGA el cual hace uso de las preferencias del DM para guiar la búsqueda en el espacio de solución, el algoritmo ayuda a mejorar la calidad de las soluciones generadas para el problema CPP.

En (González, Cervantes, Castillo, & López, 2011) se hace énfasis en la ventaja de los algoritmos multi-objetivo para la solución del problema CPP haciendo énfasis en las preferencias subjetivas del DM, haciendo uso de los métodos ELECTRE, en este artículo se hace referencia al algoritmo de NOSGA para la solución del problema CPP.

En (Fernández, n.d.) dan solución al problema CPP mediante un algoritmo genético, en cual incorporan las preferencias del DM por medio del método AHP.

Con la revisión de la literatura científica ya existen problemas que resuelven el problema de CPP que incorporan preferencias para su solución pero las implementaciones que se encontraron todas son secuenciales generando así un área de oportunidad para la investigación.

3.4. Conclusiones de la revisión de la literatura científica

En esta subsección se describen las observaciones en cuanto a la investigación y revisión de la literatura científica.

Como podemos observar en la Tabla 1 en la revisión de la literatura no se reporta el uso de estrategias paralelas que hagan uso de los métodos Electre en la solución del problema CPP el cual representa un área de oportunidad para la investigación. Por esta razón se propone desarrollar un algoritmo que haga uso de preferencias bajo los métodos Electre III y/o Electre Tri.

	Maneja Preferencias	Paralelización	Problemática
Fulop 2001	Electre Tri	NO	Desarrollar un algoritmo que reciba retroalimentación del DM
(Mousseau, Slowinski, & Zielniewicz, 2000)	Electre Tri	NO	Integración de funciones que le permiten al DM interactuar en el proceso de estimación de parámetros.
(Zak & Kruszyński, 2015)	ELECTRE III/IV	NO	Jerarquización de proyectos
(Eduardo & Carvalho, 2011)	Electre III	NO	Clasificación de Sectores Productivos
(Tam, Tong, & Lau, 2003)	Electre III	No	La selección del tipo construcción
(Nesmachnow, 2004)	N/A	MPI	Problema de Steiner Generalizado
(Rafael Arturo Trujillo Rasúa, Antonio M. Vidal Maciá, 2009)	Dominancia	OpenMP/MPI	Decodificación de señales en sistemas de comunicaciones inalámbricas
(Genis-triana & Rivera-lópez, 2010)	Dominancia	MPI	la codificación de video con balanceo predictivo
(Cort, n.d.)	N/A	OpenMP/MPI	Predicción Propagación del Fuego
(Santander-jim, Vega-rodr, Juan, & Juan, n.d.)	Dominancia	OpenMP/MPI	Inferencia Filogenética
(Ni et al., 2015)	N/A	OpenMP/MPI	Realizar simulación sísmica
(Fern & Toro, n.d.)	Dominancia	MPI	VRP
(Rodríguez et al., 2010)	Electre III	No	CPP
(González, Cervantes, Castillo, & López, 2011)	Electre III	No	CPP
(Fernández, n.d.)	AHP	No	CPP

Tabla 1.- Análisis de estado del arte.

4. Descripción del problema

En este capítulo se dará una descripción general de las características generales de las problemáticas involucradas en el proyecto de investigación, primero se describen algunas generalidades del problema CPP en el cual se dan a conocer las características del problema que son considerados para la solución del problema además de los elementos que impactan en el problema.

Posteriormente se describe algunas consideraciones que se tienen que tomar en cuenta en el proceso de incorporación de preferencias, además la importancia de incorporar preferencias en un algoritmo meta heurístico y las características de las soluciones generadas. Y finalmente se describe la importancia de paralelizar un algoritmo meta heurístico que incorpore preferencias de un tomador de decisiones.

4.1. Generalidades del Problema de cartera de Proyectos.

Los elementos mínimos que se deben de considerar para dar solución al problema de CP son los siguientes:

- El número de proyectos que compiten por el financiamiento, representada como n .
- El número de criterios considerados por el DM que ayuda a evaluar la calidad de las carteras, representada como n .
- La información relacionada con el proyecto como por ejemplo cuantos tipos de recursos se están considerando (recursos económicos, humanos o tecnológicos ente otros), en el caso particular del problema del problema CPP las restricciones presupuestarias por área y región se da a conocer el área y la región de impacto del proyecto.
- El presupuesto otorgado para el financiamiento de un cartea, se representa como C .

La complejidad del problema de optimización combinatoria CPP depende principalmente en el número de proyectos n y numero de objetivos m que intervienen en el problema, ya que a medida que n incrementa el espacio de soluciones crece exponencialmente.

Por otro lado tenemos que entre más objetivos tenga el problema entonces empieza a surgir el problema de dimensionalidad ya que encontrar las solución que mejoren dichos objetivos se vuelve cada vez más complicado.

Debido a la complejidad que con lleva solucionar un problema multi objetivo en el caso particular del problema CPP es de especial interés en el ámbito científico por lo que ha sido estudiado arduamente en literatura científica.

Una de las aportaciones que está teniendo la literatura científica es la solución del problema CPP mediante un modelo de programación lineal tri-objetivo, el cual tiene como característica principal que simula el proceso de toma de decisiones del DM.

4.2. Incorporación de Preferencias

Para comenzar con el proceso de incorporación de preferencias en un algoritmo evolutivo se tiene que tener caracterizado el proceso de Toma de Decisiones de un DM en particular, es decir que se tienen que transformar las metas de una organización, deseos, experiencias etc. en un modelo preferencial.

Una vez con teniendo el modelo preferencial de un DM se procede a incorporar la información al algoritmo meta heurístico, esta información va a fungir como mecanismo de comparación entre alternativas de solución. En este proyecto de investigación las técnicas utilizadas para la comparación entre alternativas de solución son los modelos de sobre clasificación como por ejemplo los métodos ELECTRE.

Los modelos de Sobre Clasificación tienen como característica principal que hacen uso de un grado de credibilidad para determinar si “una alternativa de solución es al menos tan buena como otra” este grado de credibilidad ayuda a generar relaciones de preferencia.

Y así identificar soluciones que se apegan fielmente al modelo preferencial de DM es decir que las soluciones generadas por el algoritmo evolutivo cubren las necesidades específicas de una organización.

Además que la incorporación de preferencias permite reducir la dimensionalidad del problema mediante un problema sub rogado de programación lineal del problema original, como por ejemplo el problema CPP.

4.3. La mejor Cartera.

Cuando al DM se le ofrecen el conjunto de soluciones no dominadas (mejores alternativas de solución) generadas por un algoritmo Meta Heurístico, entonces el DM “siente” que solo un subconjunto de estas ofrece un mejor escenario ya que estas son las que se apegan fielmente a las necesidades o las metas organizacionales. Por lo que el DM procede a identificar un sub-conjunto de soluciones basado en la premisa de que “unas carteras son mejores que otras”.

Por lo que los métodos solución para los problemas de optimización basadas en la no dominancia no son capaces de determinar por si solos cual es la “mejor Cartera” ya que no pueden identificarlas, debido a que carecen de un modelo preferencial.

Esto conlleva a que el DM tenga la responsabilidad de analizar cada una de las alternativas de solución para determinar cuáles son las soluciones que presentan un buen compromiso, lo cual representa una gran carga cognitiva para el DM.

Posteriormente de hacer este análisis tiene que hacer una comparación entre el conjunto de soluciones que tienen un buen compromiso, con la finalidad de determinar cuál es la “mejor cartera”.

Por otro lado tenemos que los métodos de solución que carecen de la información preferencial del DM estos buscan aproximar las el frente no dominado, lo cual en el proceso de búsqueda de soluciones el algoritmo puede no encontrar las soluciones que presenten los mejores escenarios para la empresa ya que la búsqueda se basa en mejorar los beneficios sin tomar en cuenta el modelo preferencial.

Abordando estas problemáticas en Fernández et al (2010) se propuso un modelo de sobre clasificación difusa, basadas en las relaciones de preferencia descritas por ROY (1996), la

característica principal de este modelo es que la calidad de la solución de una cartera no solo depende exclusivamente de la no dominancia sino que también se toman en cuenta las preferencias de un decisor mediante las relaciones de preferencia.

Con este modelo es posible identificar una sub región del frente no dominado en el cual es más probable encontrar la mejor cartera según la relación de la preferencia estricta.

Posteriormente en Fernández et. al. (2011) le hace una extensión a su modelo, agregando la preferencia débil y la k-preferencia, dándole al modelo una mayor capacidad de identificación discriminatoria.

Debido a que este modelo se basa en relaciones de sobre clasificación es indispensable calcular el grado de credibilidad de la afirmación “ a es al menos tan buena como b ” lo cual se puede calcular por los siguientes métodos reportados en la literatura:

- ELECTRE (Figueira et al., 2005; Roy, 1990) y
- PROMETHEE (Brans & Mareschal, 1990, 2005; Brans et al., 1986).

Como resultado del modelo de sobre clasificación se obtiene un modelo de programación lineal tri objetivo que además soluciona el problema del problema original. Lo cual representa un mayor coste computacional por lo que la paralización de esta meta heurística para disminuir los tiempos de ejecución.

5. Metodología de Desarrollo.

Para el desarrollo de este proyecto se propone una metodología que se encuentra dividida en cuatro etapas como se muestra en la Figura 10, posteriormente se presentan las actividades asociadas las etapas.

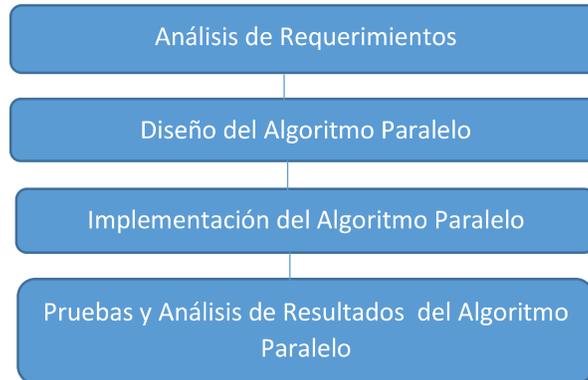


Figura 10.-Metodología de desarrollo.

En la etapa de análisis de requerimientos, se analizarán los elementos involucrados con el desarrollo de este proyecto el cual se divide en las siguientes actividades:

- Análisis de Estrategias de Incorporación de Preferencias del DM en el Problema CPP.
- Estudio de Modelos de Optimización para CPP basados en funciones de hasta 3 objetivos.
- Análisis de Algoritmos Evolutivos para CPP.
- Análisis de patrones de Diseño paralelos aplicables a CPP.

En el diseño del algoritmo paralelo se desarrollará el algoritmo basado en lo investigado en la fase anterior. Posteriormente en la implementación del algoritmo se implementará el algoritmo propuesto.

En la fase de pruebas y análisis de resultados se harán las pruebas necesarias para verificar el funcionamiento y rendimiento del algoritmo. Finalmente se evaluarán y compararán los resultados obtenidos.

6. Propuesta de solución.

Esta tesis tiene como objetivo principal desarrollar un algoritmo paralelo que incorpore preferencias para la solución del problema CPP, con la finalidad de reducir tiempos de cómputo sin que las soluciones generadas sufran una disminución en su calidad. Para lo cual se requieren realizar las siguientes actividades.

- Implementar algoritmos Meta heurísticos.
- Análisis de los algoritmos Meta heurístico implementados.
- Implementar de algoritmos Meta heurísticos paralelos.
- Análisis de los algoritmos Meta heurísticos paralelos.

La actividad de implementar algoritmos Meta heurísticos tiene como finalidad determinar cuáles son los algoritmos a implementar para posteriormente ser analizados, la actividad de análisis de los algoritmos implementados tiene como finalidad determinar cuál es el algoritmo que tuvo un mejor rendimiento para posteriormente ser implementado haciendo uso de programación paralela.

La actividad de implementar algoritmos Meta heurísticos tiene como finalidad hacer diversas implementaciones de la meta heurística que tuvo mejor rendimiento de manera secuencial para posteriormente ser analizados.

6.1. Implementación de Algoritmos Meta heurísticos.

Esta actividad tiene como finalidad la implementación de dos Meta heurísticas en su versión con incorporación de preferencias y sin incorporación de preferencias, lo cual nos ayudara a determinar cuál es el algoritmo que obtuvo un mejor rendimiento para posteriormente ser paralelizado.

Una de las meta heurísticas aplicadas son: a) GA en su versión con preferencias NOSGA-II y su versión sin preferencias NSGA-II; b) ACO, En su versión con preferencias NO-ACO, y en la versión sin preferencias NS-ACO.

6.1.1. Algoritmos Genéticos

Esta subsección tiene como finalidad dar una descripción general de los algoritmos GA implementados, así como también los parámetros de configuración de los operadores genéticos utilizados para la implementación del GA.

6.1.2. Algoritmo NSGA-II.

El algoritmo genético de ordenamiento no dominado (NSGA) propuesto por Srinivas y Deb en 1994 ha sido aplicado a varios problemas. Sin embargo debido a diversas críticas como la falta de elitismo y diversidad, en (Victoria Ruiz Martínez, 2016) se presenta un nuevo enfoque del NSGA con el fin de superar esas dificultades, además de que la complejidad del NSGA original era de $O(mn^3)$.

En el algoritmo mostrado en la Figura 11 se puede observar que se construye una población de individuos, clasificados y ordenados de acuerdo a su nivel de no dominancia. Se aplican una serie de operadores evolutivos para crear un nuevo grupo de descendientes y después se combinan los padres y la descendencia antes de dividir el nuevo grupo en frentes.

```

1    $R_t = P_t \cup Q_t$ 
2    $F = \text{fast-nondominated-sort}(R_t)$ 
3   mientras  $|P_{t+1}| < N$  hacer
4      $\text{crowding-distance-assignment}(F_i)$ 
5      $P_{t+1} = P_{t+1} \cup F_i$ 
6    $\text{Sort}(P_{t+1}, \geq_n)$ 
7    $P_{t+1} = P_{t+1}[0:N]$ 
8    $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ 
9    $t = t + 1$ 

```

Figura 11.- Algoritmo NSGA-II.

Para dividir el conjunto de la población se emplea el algoritmo *Fast Non Dominated Sorting* y se agrega una distancia a cada miembro de la población. Se emplea la distancia de *Crowding* en el operador de selección para mantener un frente diverso. Esto mantiene una mayor diversidad en la población y ayuda a que el algoritmo tenga la capacidad de hacer una mayor exploración en el espacio de soluciones.

6.1.3. Algoritmo NOSGA-II.

El algoritmo NOSGA-II fue propuesto por Fernández et. al 2013 el cual tiene como principal aportación hacer uso de relaciones de preferencia para generar un modelo de programación lineal Tri objetivo el cual busca encontrar buenas soluciones que se apeguen fielmente a las necesidades de la organización. El algoritmo se muestra en la Figura

```
PROCEDURE NOSGA II (L, Number_of_Generations)
Initialize Population P
Generate random population with size L
Evaluate objective values
Evaluate  $\sigma$  on  $P \times P$ 
For each  $x \in P$ , calculate card ( $S_{x_i}$ )
Generate fronts of equal values of card ( $S_{x_i}$ )
Assign to these fronts a rank (level) based on card ( $S_{x_i}$ )
FOR each rank  $C_i$  DO
    for each  $x \in C_i$  calculate card ( $W_{x_i}$ ), card ( $F_{x_i}$ ),  $\eta$ 
End FOR
Generate Offspring Population Q with size L
    Perform Binary Tournament Selection
    Perform Recombination and mutation
FOR  $t = 1$  to Number_of_Generations DO
    Assign  $P = P \cup Q$ 
    Evaluate  $\sigma$  on  $P \times P$ 
    FOR each parent and child in P DO
        Calculate card ( $S_{x_i}$ )
        Assign rank (level) based on card ( $S_{x_i}$ )
        Calculate  $\eta$ 
        Loop (inside) by adding solutions to the
        next generation, until L individuals have been found
    End FOR
    Replace P by the L individuals found
    Generate Offspring Population Q with size L
        Perform Binary Tournament Selection
        Perform Recombination and mutation
    End FOR
End PROCEDURE
```

Figura 12.- Algoritmo NOSGA II.

Para la implementación de este algoritmo se le hicieron se utilizaron otros operadores genéticos diferentes a los propuestos originalmente propuesto por Fernández et. al 2013, estas características se dan a conocer en la siguiente subsección.

6.1.4. Características de los GA's implementados.

Los GA's cuentan con las siguientes características: a) Selección Aleatoria; b) Cruza de n puntos; c) Mutación; la selección aleatoria es realizada solo sobre el conjunto de mejores soluciones (frente cero). En el caso del algoritmo NSGA-II esta selección se hace entre las soluciones que se encuentran en el Frente No Dominado, por otro lado tenemos que el algoritmo NOSGA-II esta selección se hace sobre las soluciones que permanecieron en el Frente No Superado Estrictamente.

El operador genético de cruce depende de las características de la instancia de manera específica en el número de proyectos involucrados por lo que la definición $n = \lfloor \text{numeroProyectos} / 2 \rfloor$, donde n representa en número de alelos de la solución que serán tomados de los padres para crear una nueva solución. Finalmente el operador de mutación utiliza la técnica de la negación de 1 bit de la solución, el bit seleccionado de manera aleatoria. Esta técnica solo es aplicada a los hijos generados y dicha mutación solo se hace a los nuevos individuos de la población (hijos) la selección de los hijos mutados dependen de una probabilidad.

6.1.5. Algoritmo de Optimización por Colonia de Hormigas.

El algoritmo implementado para resolver el problema de CPP se basó en el algoritmo propuesto por Dorigo. Se le realizó modificaciones a la actualización de feromonas (Evaporación e Intensificación) debido a que la propuesta de era para la solución del problema mono objetivo TSP.

```

ACO
Entrada numeroHormigas, problema
Salida vectorSoluciones
Inicializar Colonia(numeroHormigas, tablaFeromonas)
for i = 0 to numeroDeIteraciones
  ConstruccionSolucion(hormigai)
  busquedaLocal(solucion)
  actualizarFeromonas
end for

```

Figura 13.-Algoritmo ACO.

En la Figura 13 se muestran los componentes básicos con los que debe contar un algoritmo ACO la diferencia que radica entre esta familia de algoritmos es la evaluación de la calidad de la solución. En este proyecto de tesis la función de evaluación hizo uso del Ordenamiento No Dominado y el otro algoritmo hace uso del modelo de sobre clasificación propuesto por Fernández.

El algoritmo NS-ACO hace uso del ordenamiento no dominado para determinar la calidad de la solución, este algoritmo hace uso del Algoritmo de NON-Dominated Sorting para agrupar las soluciones por frente y para el ordenamiento se calcula la densidad de las soluciones mediante la distancia Crowding para determinar las mejores soluciones y así actualizar la tabla de feromonas.

Por otro en el algoritmo NO-ACO hace uso del método de sobre clasificación propuesto por Fernández para evaluar la calidad de la solución con la finalidad de identificar la región de soluciones prometedoras para el DM de la misma manera NOSGA-II.

6.1.6. Características de los algoritmos ACO.

Para la implementación del algoritmo ACO con y sin incorporación de preferencias en la cual se utilizó una tabla de feromonas global, además para la actualización de feromonas mediante la técnica de **Ant Colony System(ACS)** Dorigo y Gambardella (1997).

En la técnica de ACS tiene como finalidad solo darle privilegios solo a las soluciones más prometedoras encontradas por el algoritmo. Por lo que el algoritmo NS-ACO solo las hormigas que encontraron las soluciones en el frente no dominado.

Por otro lado tenemos que para el algoritmo NOSGA-II la técnica ACS le permite a las hormigas que encontraron buenas soluciones actualizar la tabla de feromonas, estas soluciones que se encuentran el ordenamiento no superado estrictamente.

En esta implementación ACO hace uso de la regla de decisión proporcional para construir la solución. La evaporación de feromonas está dada por la siguiente ecuación:

$$\tau_{ij} \leftarrow (1 - \rho) * \tau_{ij} \quad \forall i, j$$

Donde ρ representa la tasa de evaporación la cual fue definida previamente. Por otro lado tenemos que para el depósito de feromonas se hace uso de la siguiente ecuación:

$$\tau_{ij} = \begin{cases} \tau_{ij} + \Delta\tau & \text{si } x \in f_0 \\ \tau_{ij} & \end{cases}$$

Donde Δ representa la tasa de depósito de feromonas la cual se define previamente. Tanto como Δ y ρ son parámetros de configuración los cuales se mantienen constantes durante la ejecución del algoritmo. Una vez implementados los algoritmos se procede a evaluar la calidad de solución.

6.2. Características de la implementación de los Algoritmos paralelos.

Para la implementación de los algoritmos paralelos se realizara mediante el uso de patrones de diseño, la Meta heurística a paralelizar será la que se determine que tenga el mejor comportamiento en su implementación secuencial. Para la implementación de estos algoritmos se hacer mediante las siguientes arquitecturas:

- Memoria Compartida.
- Memoria Distribuida.
- Híbrida.

Las implementaciones basadas en memoria compartida se realizara mediante el uso de la librería OMP la cual está disponible para los lenguajes C/C++, ya que estas librerías están teniendo gran aceptación en la literatura científica.

Por otro lado tenemos que las implementaciones basadas en esta arquitectura se realizaran mediante el uso de la librería MPI la cual también se encuentra disponible para lenguaje C/C++, esta librería también está teniendo buena aceptación en la literatura científica.

Y finalmente para la implementación del algoritmo que haga uso de una arquitectura entre memoria compartida y memoria distribuida se hará con las librerías OMP y MPI con la finalidad de reducir tiempos de ejecución.

Todas las implementaciones paralelas tiene como finalidad disminuir los tiempo de ejecución sin perder de vista la calidad de solución en términos de la no dominancia y de la sobre clasificación. Los patrones de diseños aplicados para la programación paralela son los siguientes:

- Flujo de Datos.
- Grano Grueso.
- Maestro Esclavo.

La implementación de estos patrones será evaluados posteriormente en términos de tiempo ejecución y en términos de la calidad de Solución con la finalidad de identificar el algoritmo que tuvo mejor rendimiento.

7. Validación Experimental.

Este capítulo tiene como finalidad dar a conocer los experimentos realizados encausados a proponer un buen algoritmo paralelo haciendo uso de la información preferencial del DM, para identificar cual es el mejor algoritmo paralelo. Por lo que los experimentos realizados se dividirán las siguientes tres las siguientes etapas.

- Análisis de los Algoritmos Meta heurísticos secuenciales.
- Implementación de los algoritmos paralelos.
- Análisis de los algoritmos Paralelos.

En la etapa del análisis de los algoritmos Meta heurísticos tiene como finalidad identificar cual es la meta heurística que tuvo un mejor comportamiento para ser implementadas posteriormente en su versión paralela, después en la fase de implementación de los algoritmos paralelos tiene como finalidad dar a conocer de manera detallada la aplicación de cada uno de los patrones de diseño utilizados para la implementación paralela.

Y finalmente en la etapa de análisis de los algoritmos paralelos tiene como finalidad identificar cual es el algoritmo paralelo que tiene el mejor comportamiento, cada una de las actividades descritas anteriormente se abordan con más detalles a lo largo de esta sección. Cada uno de estos experimentos se describe a lo largo de esta subsección

7.1. Análisis de los Algoritmos Meta heurísticos secuenciales.

Este análisis tiene como finalidad hacer un análisis de cada uno de los algoritmos implementados haciendo un análisis del frente no dominado, cabe mencionar que los algoritmos implementados con incorporación de preferencia también fueron evaluados en términos de la no Dominancia para que ambos algoritmos se comparen en igualdad de condiciones.

Las características analizadas de esta fase del análisis son las siguientes: a) Comportamiento de los individuos en el frente no Dominado; b) el comportamiento de los frentes generados por el algoritmo. Para el análisis del comportamiento de los individuos en el frente no dominado se hicieron uso de los siguientes indicadores:

- **Min.** Este indicador representa la incidencia mínima de las soluciones en el frente no dominado.
- **Max.** Este indicador representa la incidencia máxima de las soluciones en el frente no dominado.
- **Promedio.** Este indicador representa la incidencia promedio de las soluciones en el frente no Dominado.
- **Desviación Estándar.** Este indicador representa que tan dispersas se encuentran las soluciones en el frente no Dominado.

Continuando con el análisis de los algoritmos Meta heurísticos secuenciales se procede a identificar cual es el algoritmo que se ve más beneficiado por la incorporación de preferencias para lo cual se hace una comparación entre el desempeño de las meta heurísticas implementadas. Esta comparación se realiza entre el algoritmo NSGA-II y el algoritmo NOSGA-II, también esta comparación es realizada entre los algoritmos NO-ACO y NS-ACO.

El propósito de estas comparaciones es observar, bajo un entorno controlado, si existen mejoras significativas al incorporar preferencias a un algoritmo evolutivos en términos de calidad de solución.

Por esta razón, los indicadores de desempeño se basaron tanto en la no dominancia como en la no superación estricta de las soluciones obtenidas.

Para la realización de este análisis se resolvieron 23 instancias del CPP, de las cuales, 20 instancias son de 4 objetivos y 25 proyectos y el resto son de 9 objetivos con 100 proyectos. Cada instancia fue resuelta por cada algoritmo 31 veces.

Este análisis fue realizado en dos etapas. La primera etapa consistió en evaluar la no dominancia existente en las soluciones generadas por los algoritmos ACO, NOACO, NSGAI y NOSGAI.

Para ello las soluciones generadas en cada ejecución, y correspondientes a una misma instancia, se integraron en dos conjuntos, un conjunto contenía las soluciones generadas por los algoritmos Genéticos NOSGAI y NSGAI, en el otro conjunto se almacenaron las soluciones generadas por los algoritmos ACO y NO-ACO. Este proceso se muestra en la Figura 14.

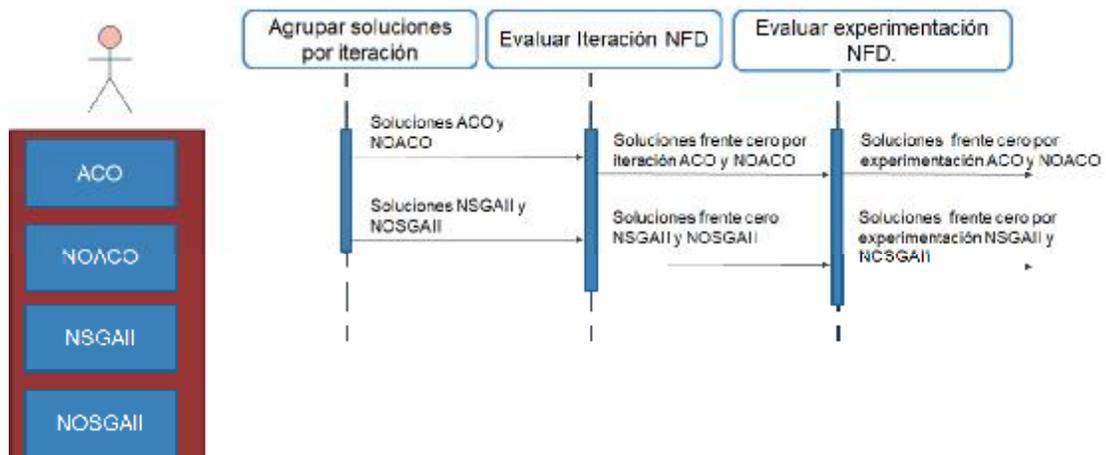


Figura 14.- Procedimiento para evaluación del frente no Dominado.

Para La evaluación de las soluciones generadas en cada ejecución, y correspondientes a una misma instancia, se integraron en dos conjuntos, un conjunto contenía las soluciones generadas por los algoritmos Genéticos NOSGAI y NSGAI así como se muestra en la Figura 14, en el otro conjunto se almacenaron las soluciones generadas por los algoritmos ACO y NO-ACO.

Una vez que las soluciones estaban agrupadas se procedió a evaluar la dominancia por instancia mediante el método *Non Fast Dominated*(Deb, Agrawal, Pratap, & Meyarivan, n.d.; Victoria Ruiz Martínez, 2016), este proceso se muestra en la Figura 15.

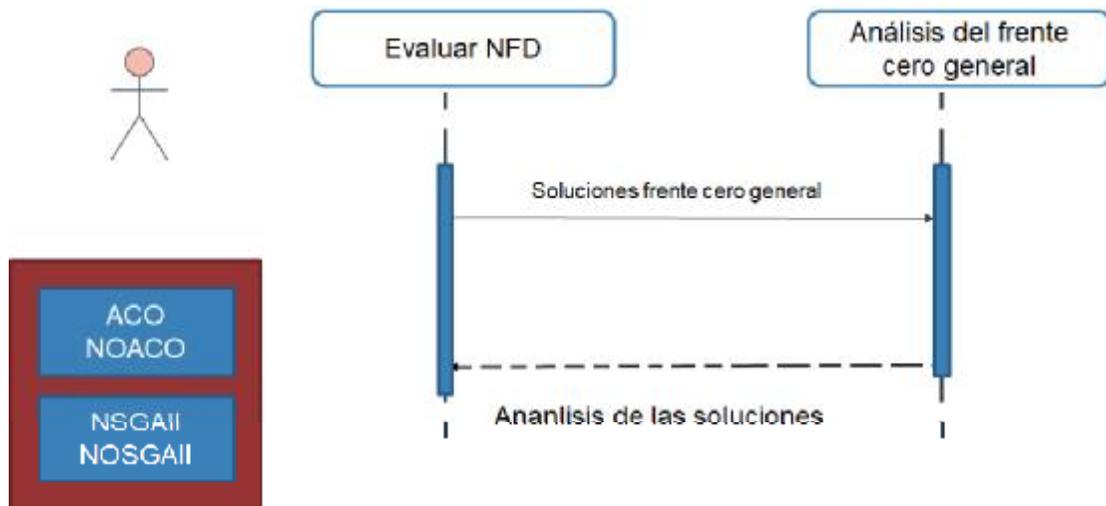


Figura 15.- Calculo del frente no dominado.

Una vez que las soluciones fueron integradas para obtener un frente no dominado por instancia, se realizó un análisis de este frente haciendo uso de los siguientes indicadores:

- *Soluciones generadas*: este indicador representa la cantidad las soluciones generadas.

- *Frente cero resultante*: este indicador representa el número de soluciones que permanecieron en el frente cero, el cual se obtuvo durante todas las experimentaciones de cada una de las instancias.
- *Porcentaje de soluciones*: este indicador presenta el porcentaje de las soluciones que permanecieron en el frente cero final.

En la segunda etapa del análisis de los algoritmos evolutivos se procedió a evaluar la frontera no superada estrictamente.

El objetivo de esta experimentación es generar la frontera no superada estrictamente sobre las soluciones que permanecieron en cada uno de los frentes no dominados en cada una de las instancias resueltas y analizar el comportamiento de las soluciones que permanecieron en esta frontera. Este proceso se muestra en la Figura 16.

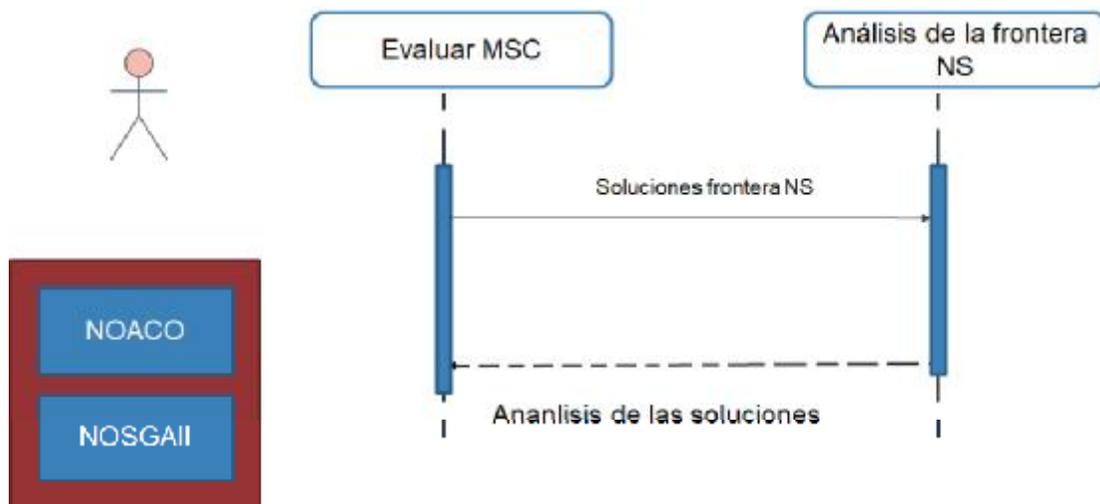


Figura 16.-Análisis del Modelo de Sobre-clasificación.

Como se puede observar en la Figura 16 Para realizar este análisis se partió de las soluciones del frente no dominado obtenido previamente. Dichas soluciones fueron evaluadas mediante el modelo de sobre clasificación difusa con la finalidad de obtener la frontera no superada estrictamente, para la realización del análisis de esta frontera se hace uso de los siguientes indicadores:

- *Soluciones generadas*: Este indicador representa las soluciones que permanecieron en el frente cero por en las experimentaciones realizadas.
- *Porcentaje de soluciones repetidas*: Este indicador representa el porcentaje de soluciones repetidas por algoritmo.
- *Soluciones no repetidas*: Este indicador representa el número de soluciones no repetidas en el frente cero por iteración.
- *Soluciones NS*: Este indicador representa las soluciones que quedaron en la frontera no superada estrictamente.

El propósito de estos análisis es evaluar el impacto que obtiene un algoritmo al incorporarle preferencias, ya que el objetivo que se persigue con esta experimentación es determinar cuál es el algoritmo que tuvo un mejor comportamiento.

Este comportamiento depende no solo de que las soluciones tengan una buena calidad de solución en términos de la no dominancia, sí que además estas soluciones se apeguen al modelo preferencial de un decisor.

Lo cual ayudara a identificar cual fue el mejor algoritmo y que posteriormente mediante el uso de la programación paralela contribuirá a disminuir el tiempo de ejecución del algoritmo que tuvo el mejor comportamiento.

7.2. Resultados derivados de la Experimentación.

Para iniciar con el análisis de los algoritmos implementados se realizará el análisis individual de los frentes no dominados generados a lo largo de las iteraciones generadas durante la ejecución del algoritmo, se iniciara con el algoritmo NS-ACO.

Este análisis iniciara con el del comportamiento de los individuos en el frente no dominado por iteracion, estos resultados obtenidos de la experimentación se muestran en la Tabla 3.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	7	34	17.1029651	0.23263142
o4p25_1	8	36	17.0746171	0.37145587
o4p25_2	9	39	18.826002	0.4315585
o4p25_3	9	35	16.6311502	0.27632931
o4p25_4	5	28	12.1163245	0.30626627
o4p25_5	13	43	21.6695992	0.40031105
o4p25_6	11	35	19.4369501	0.32105535
o4p25_7	7	31	13.7885305	0.33649371
o4p25_8	7	31	13.7885305	0.33649371
o4p25_9	5	37	11.7771261	0.35807497
o4p25_10	11	37	19.8403389	0.31468975
o4p25_11	4	23	9.99576409	0.2393195
o4p25_12	6	30	13.3343109	0.27114211
o4p25_13	3	19	7.54936461	0.19714566
o4p25_14	12	36	19.7504073	0.37473055
o4p25_15	8	36	16.935158	0.42016853
o4p25_16	9	28	15.1746497	0.26359938
o4p25_17	9	34	18.2130987	0.32117046
o4p25_18	7	34	15.2574128	0.30167546
o4p25_19	11	40	19.8230694	0.33323638
o4p25_20	8	31	15.2372108	0.29344993
o9p100_0	30	79	49.4822418	0.31221256
o9p100_1	52	104	80.1564027	2.91435779
o9p100_2	12	97	75.7454839	2.83925091

Tabla 2.- Comportamiento de individuos en el no dominado del algoritmo NS-ACO.

Como se puede observar en la Tabla 2 en las instancias de 9 objetivos con 100 proyectos se encontraron un mayor número de soluciones en el frente no dominado por iteración en comparación con las instancias de 4 objetivos con 100 proyectos.

Ya que en las instancias de 9 objetivos con 100 proyectos el promedio máximo alcanzado fue de 80 soluciones en el frente no domina en la *o9p100_1*. Por otro lado tenemos que la instancia *o9p100_0* fue la que obtuvo un promedio de soluciones en el frente no dominado más bajo con 40 soluciones en este frente.

En términos generales podemos decir que el índice de dispersión del número de soluciones que permanecieron en el frente no dominado (Desviación estándar) por iteración en las instancias de tienen 9 objetivos y 100 proyectos fue mayor que en las instancias de 4 objetivo con 25 proyectos, ya que las instancias de 9 objetivos al menos duplican el grado de dispersión comparada con la instancia de 4 objetivos y 100 proyectos.

Continuando con el análisis de la Meta heurística NS-ACO, se analizara el número de frentes generados en el transcurso de las iteraciones de este algoritmo, estos resultados se muestran en la Tabla 4.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	4	8	5.42196155	0.05913871
o4p25_1	4	8	5.80579993	0.05535446
o4p25_2	4	8	5.47344412	0.05114854
o4p25_3	4	8	5.78136201	0.0566741
o4p25_4	4	9	6.03812317	0.04712103
o4p25_5	4	8	5.39622027	0.04611194
o4p25_6	4	9	5.60410557	0.05392859
o4p25_7	4	10	6.34115347	0.04996384
o4p25_8	4	10	6.34115347	0.04996384
o4p25_9	5	9	6.63668948	0.06811203
o4p25_10	3	8	5.23232323	0.05799682
o4p25_11	3	8	5.29749104	0.05431268
o4p25_12	4	9	6.23525578	0.04278045
o4p25_13	2	6	3.71293581	0.06552215
o4p25_14	4	9	5.63180189	0.05587942
o4p25_15	4	9	5.91951776	0.0538762
o4p25_16	4	9	6.13554904	0.05811325
o4p25_17	4	8	5.48517432	0.04899616
o4p25_18	4	9	6.19354839	0.05409856
o4p25_19	4	9	5.66177908	0.04529325
o4p25_20	4	8	5.60084718	0.04689145
o9p100_0	2	5	3.24666015	0.02200909
o9p100_1	2	2	2	0
o9p100_2	2	2	2	0

Tabla 3.- Comportamiento de los frentes del algoritmo NS-ACO.

Como podemos observar en la Tabla 3 en dos instancias de 9 proyectos con 100 objetivos se mantuvo sin variación ya que se generaron 2 frentes en cada una de las iteraciones realizadas a las instancias *o9p100_1* y *o9p100_1*.

También se puede observar que en la instancia *o9p100_0* la incidencia de mínima de frentes generados fueron 2, comparando esto con el promedio de las otras 2 instancias de 9 objetivos es la misma incidencia. Por lo que en esta instancia existe una mayor variabilidad en el número de frentes generados.

Además se puede observar que en las instancias de 4 objetivos con 25 proyectos se puede observar que en el 80% de las instancias presentó una incidencia mínima de 4 frentes generados a lo largo de las iteraciones realizadas por el algoritmo, por otro lado tenemos que en el 43 % de las instancias presentó que en las experimentaciones se obtuvieron 8 frentes.

Por lo que en el otro 43% de las instancias resueltas se observó que en algunas iteraciones se generaron a lo más 9 frentes durante las experimentaciones realizadas, también se observó que en solo dos soluciones se generaron a lo más 10 frentes en algunas experimentaciones realizadas.

Por otro lado tenemos que en el indicador promedio se reportan que solo en una instancia se obtuvieron 3 frentes en promedio y en el resto de las instancias el número de frentes osciló entre 5 y 6 frentes generados en las iteraciones generadas por el algoritmo. Además se puede observar que en las instancias de 4 objetivos con 25 proyectos se obtuvo una desviación estándar promedio de 0.05339417. La siguiente Meta heurística a analizar es la Meta heurística NSGA-II, este análisis iniciará con el comportamiento de los individuos del frente no Dominado derivados de la experimentación realizada se muestran en la Tabla 5.

	Mínimo	Máximo	Promedio	Desviación Estándar
op25_0	30	73	48.3311	3.02737
o4p25_1	30	69	32.216	8.77033
o4p25_2	30	69	32.216	8.77033
o4p25_3	30	69	32.216	8.77033
o4p25_4	30	69	32.216	8.77033
o4p25_5	30	69	32.216	8.77033
o4p25_7	30	69	32.216	8.77033
o4p25_8	30	69	32.216	8.77033
o4p25_9	30	69	32.216	8.77033
o4p25_10	12	75	34.0971	7.3355
o4p25_11	6	71	30.7071	6.71197
o4p25_12	5	71	40.7113	3.55869
o4p25_13	7	69	42.5823	5.01229
o4p25_14	8	70	35.8384	7.57563
o4p25_15	31	67	31.998	8.58388
o4p25_16	20	65	40.1065	3.51603
o4p25_17	25	78	54.6406	2.54778
o4p25_18	22	74	38.5937	5.68167
o4p25_19	31	82	45.8501	4.55623
o4p25_20	35	70	38.8296	5.57622

o9p100_0	53	100	78.9003	3.10785
o9p100_1	52	104	80.1564	2.91436
o9p100_2	12	97	75.7455	2.83925

Tabla 4.- Comportamiento de los individuos en el no dominado del algoritmo NSGA-II

Como se puede observar en la Tabla 5, en 2 instancias de 9 objetivos se obtuvieron que la incidencia mínima fue mayor que en las instancias de 4 objetivos con 25 proyectos, y solo en la instancia *o9p100_2* solo se obtuvo 12 que al menos se obtuvieron 12 soluciones en el frente no dominado por iteración.

Por otro lado tenemos que en la incidencia máxima de las soluciones en el frente no dominado por iteración en las instancias 9 objetivos y 100 proyectos oscilo entre 97 y 104 soluciones. Por otro lado tenemos que se obtuvo una incidencia promedio que oscila entre 75 y 80 soluciones en el frente no dominado.

Además se observó que en las soluciones de 4 objetivos con 25 proyectos se obtuvo que en el 47% de las instancias se alcanzó una incidencia mínima de 30 soluciones en el frente no dominado. Además se observa que en una solo instancia se obtuvo 35 soluciones en el frente no dominado. Por lo que se observa que en la incidencia máxima a lo más se observa que existen 69 soluciones en el frente no dominado por iteración.

También se puede observar que en el indicador de incidencia promedio en las instancias de 4 objetivos y 25 proyectos con al menos 31 soluciones en el frente no dominado, por otro lado tenemos que en estas también existió una mayor incidencia que en las instancias de 9 objetivos con 100 proyectos.

Continuando con el análisis del algoritmo NSGA-II se procede a evaluar el comportamiento del comportamiento de los frentes generados por este algoritmo, los resultados obtenidos se muestran en la Tabla 6.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	2	2	2	0
o4p25_1	2	2	2	0
o4p25_2	2	2	2	0
o4p25_3	2	2	2	0
o4p25_4	2	2	2	0
o4p25_5	2	2	2	0
o4p25_6	2	2	2	0
o4p25_7	2	2	2	0
o4p25_8	2	2	2	0
o4p25_9	2	2	2	0
o4p25_10	2	2	2	0
o4p25_11	2	2	2	0
o4p25_12	2	2	2	0
o4p25_13	2	2	2	0
o4p25_14	2	2	2	0
o4p25_15	2	2	2	0
o4p25_16	2	2	2	0
o4p25_17	2	2	2	0

o4p25_18	2	2	2	0
o4p25_19	2	2	2	0
o4p25_20	2	2	2	0
o9p100_0	2	2	2	0
o9p100_1	2	2	2	0
o9p100_2	2	2	2	0

Tabla 5.- Comportamiento de los frentes del algoritmo NSGA-II

Como se puede observar en la Tabla 6 se observó que a lo largo de todas las iteraciones del algoritmo durante las experimentaciones realizadas a las instancias se observó que el algoritmo se mantuvo generando 2 frentes en cada una de las de sus iteraciones.

Por lo que se puede observar entre los algoritmos propuestos sin incorporación de preferencias (NS-ACO, NSGA-II) el algoritmo que tiene una mayor variabilidad en la generación de frentes el algoritmo NS-ACO.

Por otro lado tenemos que en solo una de las instancias de 9 objetivos con 100 proyectos el algoritmo NSGA-II supera al algoritmo NS-ACO. Además tenemos que en las instancias de 4 objetivos con 25 proyectos supera el algoritmo NSGA-II supera al algoritmo NS-ACO.

Continuando con este análisis se procede a analizar los algoritmos que incorporan preferencias, este análisis iniciara con la Meta heurística a analizar es NOSGA-II. Este análisis iniciara con el comportamiento de los individuos de los individuos en el frente no Dominado los resultados derivados de esta experimentación se muestran en la Tabla 7.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	12	97	75.7454839	2.83925091
o4p25_1	21	71	34.6471163	6.74584739
o4p25_2	23	77	49.5027696	2.46683874
o4p25_3	18	67	45.8041707	3.05086978
o4p25_4	24	73	53.2564353	2.10519109
o4p25_5	22	75	52.0087977	2.23505434
o4p25_7	22	22	13.3225806	0.21872376
o4p25_8	21	72	48.3681981	2.44346905
o4p25_9	16	69	45.3688498	3.68499551
o4p25_10	21	70	31.1756272	10.5385932
o4p25_11	24	71	30.7070707	6.71196824
o4p25_12	27	71	40.7113066	3.55868826
o4p25_13	30	69	42.5822744	5.01229096
o4p25_14	34	70	35.8383838	7.57563421
o4p25_15	22	72	29.659824	9.23330832
o4p25_16	16	68	43.9156077	1.89012189
o4p25_17	16	68	43.9156077	1.89012189
o4p25_18	22	74	38.5936787	5.68167012
O4p25_19	23	68	44.1423917	2.14750062
O4p25_20	20	68	41.8752036	1.95741778
O9p100_0	16	64	49.8106875	2.63170186

O9p100_1	19	65	51.0928641	2.11262359
O9p100_2	17	64	46.7869013	2.4435499

Tabla 6.- Comportamiento de individuos en el no dominado del algoritmo NOSGA-II

Como se puede observar en la Tabla 7 las instancias de instancias de 9 objetivos en dos instancias se obtuvo una menor incidencia mínima de soluciones en el frente no dominado comparado con los algoritmos sin articulación de preferencias ya que en solo la estancia o9p100_2 obtiene una incidencia mínima mayor. También se observa que en las instancias de 4 objetivos con 100 proyectos su incidencia mínima es menor que con el algoritmo genético que no incorpora preferencias, por otro lado tenemos que el algoritmo NOSGA-II tiene una incidencia máxima mayor comparado con el NSGA-II ya que el algoritmo NOSGA-II oscila entre 68 y 97 soluciones en este indicador.

Además se observa que en el indicador promedio el algoritmo NOSGA-II supera al algoritmo NSGAII en un 52% en las instancias de 4 objetivos con 25 proyectos, pero en las instancias de 9 objetivos con 100 proyectos el algoritmo NSGA-II tiene un mejor comportamiento que el NOSGA-II.

Continuando con el análisis de la Meta heurística NOSGA-II, se procederá a analizar el comportamiento de los frentes generados por este algoritmo, los resultados derivados de esta experimentación se muestran en la Tabla 8.

	Mínimo	Máximo	Promedio	Desviación Estándar	
o4p25_0	2	2	2	0	
o4p25_1	2	2	2	0	
o4p25_2	2	2	2	0	
o4p25_3	2	2	2	0	
o4p25_4	2	2	2	0	
o4p25_5	2	2	2	0	
o4p25_6	2	2	2	0	
o4p25_7	2	2	2	0	
o4p25_8	2	2	2	0	
o4p25_9	2	2	2	0	
o4p25_10	2	2	2	0	
o4p25_11	2	2	2	0	
o4p25_12	2	2	2	0	
o4p25_13	2	2	2	0	
o4p25_14	2	2	2	0	
o4p25_15	2	2	2	0	
o4p25_16	2	2	2	0	
o4p25_17	2	2	2	0	
o4p25_18	2	2	2	0	
o4p25_19	2	2	2	0	
o4p25_20	2	2	2	0	
o9p100_0	2	2	2	0	
o9p100_1	2	2	2	0	
o9p100_2	2	2	2	0	

Tabla 7.- Comportamiento de los frentes generados por el algoritmo NOSGA-II

Como se puede observar en la Tabla 8 tenemos que en términos de la generación de frentes generados en las experimentaciones tiene el mismo comportamiento que el algoritmo NSGA-II. Por lo que algoritmo que provee una mayor variabilidad en el número de frentes generados es el algoritmo NS-ACO.

La siguiente Meta heurística a analizar es la de NO-ACO, este análisis iniciara con el comportamiento de los individuos generados en el frente no Dominado, los resultados derivados de esta experimentación se muestran en la Tabla 9.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	7	42	30.8778104	0.369613832
o4p25_1	7	42	30.8778104	0.369613832
o4p25_2	11	46	32.9175627	0.259465976
o4p25_3	4	43	30.3307266	0.358066261
o4p25_4	4	45	33.1130661	0.444252983
o4p25_5	9	45	32.378625	0.260252103
o4p25_7	2	44	30.6008472	0.50204452
o4p25_8	6	42	31.2971652	0.314679572
o4p25_9	13	48	35.6927338	0.339646595
o4p25_10	1	25	10.8631476	1.221821097
o4p25_11	1	37	21.1990877	0.982450627
o4p25_12	1	38	21.2323232	1.066074445
o4p25_13	1	38	21.2323232	1.066074445
o4p25_14	4	43	29.1287064	0.546774425
o4p25_15	1	33	17.6546106	1.372663252
o4p25_16	9	46	33.3492994	0.31421643
o4p25_17	7	42	29.8246986	0.281575655
o4p25_18	2	43	30.5676116	0.574338403
o4p25_19	4	45	31.9758879	0.452812209
o4p25_20	4	33	23.1717172	0.303438416
o9p100_0	5	38	27.0811339	0.295480695
o9p100_1	5	38	26.6901271	0.455145778
o9p100_2	5	40	27.7777778	0.304049042

Tabla 8.- Comportamiento de las soluciones en el frente no dominado del algoritmo NO-ACO

Como podemos observar en la Tabla 9 podemos observar que en las soluciones de 9 objetivos con 100 proyectos en el indicador de incidencia mínima al algoritmo obtuvo una menor incidencia comparado con el algoritmo NSGA-II, por otro lado en el indicador de incidencia máxima de soluciones en frente no dominado es menor comparado con el NSGA-II así como en el indicador en el indicador promedio y desviación estándar.

Además tenemos que en las instancias de 4 objetivos con 25 proyectos solo en una instancia del algoritmo NO-ACO tiene una incidencia promedio mayor que el algoritmo NSGA-II. Pero el algoritmo NOACO tiene un menor grado de dispersión en el número de soluciones del frente no dominado.

Para terminar con esta etapa del análisis de las Meta heurísticas implementadas se analizará el algoritmo NO-ACO, por lo que iniciaremos con el análisis del comportamiento de los frentes generados por este algoritmo, estos resultados se muestran en la Tabla 10.

	Mínimo	Máximo	Promedio	Desviación Estándar
o4p25_0	4	8	5.42196155	0.05913871
o4p25_1	4	8	5.80579993	0.05535446
o4p25_2	4	8	5.47344412	0.05114854
o4p25_3	4	8	5.78136201	0.0566741
o4p25_4	4	9	6.03812317	0.04712103
o4p25_5	4	8	5.39622027	0.04611194
o4p25_6	4	9	5.60410557	0.05392859
o4p25_7	4	10	6.34115347	0.04996384
o4p25_8	4	10	6.34115347	0.04996384
o4p25_9	5	9	6.63668948	0.06811203
o4p25_10	3	8	5.23232323	0.05799682
o4p25_11	3	8	5.29749104	0.05431268
o4p25_12	4	9	6.23525578	0.04278045
o4p25_13	2	6	3.71293581	0.06552215
o4p25_14	4	9	5.63180189	0.05587942
o4p25_15	4	9	5.91951776	0.0538762
o4p25_16	4	9	6.13554904	0.05811325
o4p25_17	4	8	5.48517432	0.04899616
o4p25_18	4	9	6.19354839	0.05409856
o4p25_19	4	9	5.66177908	0.04529325
o4p25_20	4	8	5.60084718	0.04689145
o9p100_0	2	5	3.24666015	0.02200909
o9p100_1	2	2	2	0
o9p100_2	2	2	2	0

Tabla 9.- Comportamiento de los frentes generados por el algoritmo NO-ACO

Como podemos observar en la Tabla 10 el algoritmo NO-ACO este algoritmo tiene una mayor variabilidad en el número de frentes generados comparados con los algoritmos NSGAII y NOSGA-II ya que en todas las instancias de 4 objetivos con 25 proyectos presenta un comportamiento con una mayor variabilidad en comparación con los algoritmos NSGA-II y NOSGA-II.

Pero tenemos que en 2 de las 3 instancias de 9 objetivos con 100 proyectos no existe variabilidad de la misma manera que los algoritmos NSGA-II y el algoritmo NOSGA-II, además que el grado de dispersión es muy bajo.

Continuando con el análisis de los algoritmos secuenciales implementados se procederá a determinar cuál es el algoritmo que se ve más beneficiado por la incorporación de preferencias en términos de la no Dominancia y en términos del modelo de Sobre Clasificación Difusa. Este análisis iniciará con la comparación de los algoritmos NSGA-II y NOSGA-II en términos de la no dominancia, lo resultados obtenidos se muestran en la Tabla 11.

Instancia	Min		Max	
	NSGAI	NOSGAI	NSGAI	NOSGAI
o4p24_0	0.33801717	0.67777778	0.99144934	0.99879554
o4p24_1	0.15200967	0.32452431	0.9878125	0.99131244
o4p24_2	0.33655492	0.80609267	0.99700727	0.99804257
o4p24_3	0.32894126	0.73002877	0.99305409	0.99722736
o4p24_4	0.15023923	0.41499086	0.99794003	0.99966678
o4p24_5	0.32568504	0.73960857	0.99055695	0.99679601
o4p24_7	0.3463086	0.75089547	0.99935065	0.997544
o4p24_8	0.31836875	0.59511811	0.99209402	0.9983101
o4p24_9	0.3159148	0.69830508	0.9886994	0.99838083
o4p24_10	0.1586907	0.50421179	0.99733079	0.99637074
o4p24_11	0.11236559	0.41876348	0.96763422	1
o4p24_12	0.24633364	0.38451723	0.97040169	0.99778123
o4p24_13	0.32233826	0.61919505	0.99798682	0.99735527
o4p24_14	0.33514952	0.50737042	0.99825141	0.99538867
o4p24_15	0.10505451	0.15526316	0.99363671	0.99582617
o4p24_16	0.30646139	0.71366045	0.999464	0.99722126
o4p24_17	0.36849286	0.79268055	0.99110946	0.9989083
o4p24_18	0.16606498	0.40820456	0.98732326	0.9964497
o4p24_19	0.33466956	0.55000754	0.995981	0.99795849
o4p24_20	0.33548967	0.78972768	0.99494557	1
o9p100_0	0.90921872	0.77377892	0.99761297	0.99602529
o9p100_1	0.86541738	0.82917988	0.99528894	0.99564888
o9p100_2	0.82749955	0.85731678	0.99874759	0.99623777

Tabla 10.- Comparación del frente no dominado de los algoritmos Genéticos en el indicador Min y Max.

Como se puede observar en la Tabla 11 en las instancias de 4 objetivos con 25 proyectos en algoritmo NOSGA-II una mayor incidencia en el indicador min con una diferencia promedio de 0.30888967 y una varianza de 0.01279763.

Además se observó que en el algoritmo que tuvo una incidencia mayor en 2 de las 3 instancias de 9 objetivos con 100 proyectos en el indicador Min fue el algoritmo NSGA-II con una diferencia promedio de 0.08583865 y una varianza de 0.00246027 pero en la instancia *o9p100_2* el algoritmo que tuvo una incidencia mayor fue la del algoritmo NOSGA-II con una diferencia de 0.02981723.

También se observó que en el 75% de las instancias tuvo una incidencia mayor en el indicador en la instancia de 4 objetivos con 25 proyectos fue el algoritmo NOSGA-II con una diferencia promedio de 0.00838733 y una varianza de 8.42371E-05.

Por otro lado tenemos también que en el 25% de las instancias el algoritmo NSGA-II obtiene una incidencia mayor en el indicador Max con una diferencia promedio de 0.00170075 y una varianza de 6.69403E-07.

Como podemos ver el algoritmo que tuvo una diferencia mayor en el indicador Max en la instancias de 4 objetivos con 25 proyecto fue el algoritmo NOSGA-II, pero el algoritmo que tuvo una varianza menor fue el NSGA-II.

Por otro lado tenemos que en 2 de las 3 instancias de 9 objetivos con 100 proyectos el algoritmo que tuvo una mayor incidencia en el indicador Max fue el algoritmo NSGA-II con una diferencia promedio de 0.00204875, pero el algoritmo NOSGA-II en la instancia *o9p100_1* tiene una diferencia de 0.00035994.

Continuando con el análisis basado en el frente no dominado se procederá a evaluar la incidencia de las soluciones en este frente bajo los indicadores: a) Promedio; b) Varianza; estos resultados se muestran en la Tabla 12.

Instancia	Promedio		Varianza	
	NSGAI	NOSGAI	NSGAI	NOSGAI
o4p24_0	0.590582	0.89207621	0.05799605	0.01115184
o4p24_1	0.5558839	0.81097186	0.05028934	0.02900829
o4p24_2	0.73436198	0.92295285	0.04712406	0.00534473
o4p24_3	0.58243886	0.89465114	0.04981656	0.00858652
o4p24_4	0.60451344	0.91038324	0.08335584	0.02379198
o4p24_5	0.65577789	0.89002936	0.0459619	0.00847438
o4p24_7	0.70397512	0.90755895	0.05711823	0.00679907
o4p24_8	0.63789139	0.86266788	0.05527897	0.01886962
o4p24_9	0.67447821	0.88448469	0.05283604	0.01051615
o4p24_10	0.62401903	0.82761019	0.05862593	0.0300079
o4p24_11	0.57461195	0.85175539	0.04121032	0.02313764
o4p24_12	0.67495146	0.76155691	0.05242556	0.0471222
o4p24_13	0.68555976	0.91059727	0.06775984	0.01085898
o4p24_14	0.58790313	0.85211459	0.04716699	0.02241817
o4p24_15	0.51778465	0.83121918	0.04213185	0.03933736
o4p24_16	0.58141118	0.93935614	0.06124048	0.00377554
o4p24_17	0.68593642	0.94411476	0.07171715	0.00332352
o4p24_18	0.60158151	0.87371712	0.06016314	0.02873039
o4p24_19	0.64487349	0.86593782	0.06196143	0.02571321
o4p24_20	0.59725894	0.9343272	0.05345363	0.00355365
o9p100_0	0.97630868	0.96088179	0.00048302	0.00201762
o9p100_1	0.96778434	0.96442113	0.00079367	0.00162285
o9p100_2	0.96162407	0.96440088	0.00124976	0.00111113

Tabla 11.-Comparación del frente no dominado de los algoritmos Genéticos en el indicador Promedio y Varianza.

Como se puede observar en las instancias de 4 objetivos con 25 proyectos el algoritmo que tuvo un mejor desempeño en el indicador promedio es el algoritmo NOSGA-II, lo que quiere decir que la

incorporación de preferencias en este algoritmo genético le ayudo a aproximar mejor el frente con dominado ya que existió una diferencia en promedio de 0.25261442 y una varianza de 0.00365032.

Pero por otro lado tenemos que en las instancias de 9 objetivos con 100 proyectos el algoritmo NSGA-II tuvo un mejor desempeño en 2 de las 3 instancias en este indicador con una diferencia promedio de 0.00939505, donde eta diferencias es menor que en la diferencia promedio de las instancias de 4 objetivos con 25 proyectos.

Además tenemos las soluciones generadas por el algoritmo NOSGA-II tiene como característica principal que las soluciones generadas se apegan al modelo preferencial del DM lo que quiere decir que estas soluciones tienen un mejor compromiso y que a su vez el perfil de preferencia utilizado en este algoritmo también ayuda aproximar mejor el frente no dominado.

Continuando con el análisis comparativo entre los algoritmos implementados, se analizará los algoritmos ACO y NO-ACO en términos de la frontera no dominada en los indicadores Min y Max los cuales se muestran en la Tabla 13.

Instancia	Min		Max	
	NS-ACO	NOACO	NS-ACO	NOACO
o4p24_0	0.37780473	0.79279279	0.41387705	0.94957983
o4p24_1	0.39050445	0.62765957	0.42111578	0.86458333
o4p24_2	0.37369641	0.81308411	0.3994723	0.95283019
o4p24_3	0.35696203	0.8440367	0.39063502	0.97321429
o4p24_4	0.34889643	0.73333333	0.39863714	0.95412844
o4p24_5	0.3812882	0.84482759	0.40510597	0.94845361
o4p24_7	0.34744526	0.77142857	0.38629518	0.97272727
o4p24_8	0.37794844	0.81481481	0.41114058	0.9245283
o4p24_9	0.33771107	0.70652174	0.37967914	0.89380531
o4p24_10	0.40605721	0.61261261	0.44393939	0.97826087
o4p24_11	0.32212581	0.61052632	0.3960396	0.98039216
o4p24_12	0.34444444	0.76923077	0.375	0.93333333
o4p24_13	0.34444444	0.76923077	0.375	0.93333333
o4p24_14	0.38692673	0.77884615	0.42063908	0.96153846
o4p24_15	0.34572733	0.77697842	0.41274752	1
o4p24_16	0.38642473	0.71428571	0.42328398	0.90566038
o4p24_17	0.37007424	0.80555556	0.40047818	0.97196262
o4p24_18	0.35891813	0.77678571	0.39263804	0.99038462
o4p24_19	0.37604167	0.85454545	0.40138963	1
o4p24_20	0.36566643	0.62105263	0.41584158	0.86324786
o9p100_0	0.64426559	0.98019802	0.66847147	1
o9p100_1	0.65195138	0.98039216	0.67036011	1
o9p100_2	0.64454565	0.92380952	0.66323746	1

Tabla 12.-Comparación del frente no dominado de los algoritmos ACO en el indicador Min y Max.

Como se puede observar en la Tabla 13 el algoritmo que tiene un mejor comportamiento fue el algoritmo NO-ACO ya que este algoritmo supero a el algoritmo NS-ACO en todas las instancias, por

lo que en el indicador Min se obtuvo una diferencia promedio de 0.38695206 en las instancias de 4 objetivos con 25 proyectos y en las instancias de 9 objetivos con 100 proyectos obtuvo una diferencia promedio de 0.31454569.

Por otro lado tenemos que en indicador Max el algoritmo que tuvo un mejor comportamiento fue el algoritmo NO-ACO debido a que supero al algoritmo NS-ACO en todas las instancias, en las instancias de 4 objetivos con 25 proyectos se obtuvo una diferencia de 0.54445045. Por otro lado tenemos que en las instancias de 9 objetivos con 100 proyectos se alcanzó una diferencia promedio de 0.33264365.

Continuando con el análisis de los algoritmos NS-ACO y el NO-ACO se proceden a evaluar los indicadores promedio y varianza, estos resultados se muestran en la Tabla 14.

Instancia	Promedio		Varianza	
	NS-ACO	NOACO	NS-ACO	NOACO
o4p24_0	0.39537662	0.88086837	8.4417E-05	0.0016391
o4p24_1	0.40571953	0.76386787	0.00006709	0.0039434
o4p24_2	0.38822744	0.88545402	3.9251E-05	0.00158671
o4p24_3	0.36949951	0.9338079	5.8558E-05	0.00111857
o4p24_4	0.37901012	0.85317084	0.00014973	0.00247771
o4p24_5	0.39445724	0.89755215	3.6664E-05	0.00089682
o4p24_7	0.3652391	0.86725189	8.3055E-05	0.0025932
o4p24_8	0.39224434	0.87709794	4.6584E-05	0.00103657
o4p24_9	0.35793731	0.80896938	0.00010204	0.00212193
o4p24_10	0.42692835	0.84915997	6.3317E-05	0.00849062
o4p24_11	0.36602777	0.86601283	0.00040688	0.00929003
o4p24_12	0.3609783	0.84544234	7.7814E-05	0.00215974
o4p24_13	0.3609783	0.84544234	7.7814E-05	0.00215974
o4p24_14	0.40329841	0.88304666	5.1051E-05	0.00251951
o4p24_15	0.38559112	0.90640059	0.00020881	0.00374358
o4p24_16	0.40137672	0.80558289	9.4424E-05	0.00234295
o4p24_17	0.38517433	0.88939594	5.1424E-05	0.00118438
o4p24_18	0.37576601	0.89296256	6.2227E-05	0.0028176
o4p24_19	0.38928608	0.93485121	4.5672E-05	0.00103015
o4p24_20	0.39086663	0.75353879	0.000129	0.00358772
o9p100_0	0.65604985	0.99215323	0.00002153	5.1417E-05
o9p100_1	0.66217581	0.99345308	1.9788E-05	4.5127E-05
o9p100_2	0.65459228	0.99007156	2.2595E-05	0.00023475

Tabla 13.-Comparación del frente no dominado de los algoritmos ACO en el indicador Promedio y Varianza.

Como podemos observar en la Tabla 13 el algoritmo que obtuvo el mejor comportamiento en el indicador promedio fue el algoritmo NO-ACO ya que supero al algoritmo NS-ACO en todas las instancias resueltas, en promedio existió una incidencia mayor en cada una de las experimentaciones realizadas con una diferencia promedio de 0.45864144.

Por lo que tenemos que en algoritmo NO-ACO supero en los indicadores Min, Max y promedio de la incidencia por lo que el algoritmo que se vio más beneficiado al explorar el frente no dominado con la incorporación de preferencias fue el algoritmo NO-ACO.

Comparando el beneficio obtenido por el algoritmo NOSGA-II no fue tan significativo comparado con el beneficio obtenido por el algoritmo NO-ACO.

Una vez concluido el análisis de los frentes no dominados generados por iteración se procede a analizar el frente no dominado general, este análisis iniciara con los resultados obtenidos de los algoritmos NSGA-II y NOSGA-II los cuales se muestran en la Tabla 15.

Instancia	Soluciones generadas		Frente cero resultante		Porcentaje de Soluciones	
	NSGA	NOSGA	NSGA	NOSGA	NSGA	NOSGA
o4p25_0	273622	152267	32270	13413	0.11793642	0.08808869
o4p25_1	191437	93497	8648	5591	0.04517413	0.05979871
o4p25_2	282484	176448	44359	18830	0.1570319	0.10671699
o4p25_3	281392	160973	61365	34826	0.21807656	0.21634684
o4p25_4	239317	160471	21038	12382	0.08790851	0.07716036
o4p25_5	310008	163825	48124	31714	0.1552347	0.19358462
o4p25_7	249666	160504	45765	32069	0.1833049	0.19980187
o4p25_8	251077	144670	38666	28778	0.15400057	0.19892168
o4p25_9	259916	176518	9142	10570	0.0351729	0.05988058
o4p25_10	216946	129094	55936	35100	0.25783375	0.2718949
o4p25_11	214098	103139	33310	17459	0.15558296	0.16927641
o4p25_12	226932	152860	42213	21306	0.18601608	0.13938244
o4p25_13	278974	144672	42213	21306	0.15131518	0.14727107
o4p25_14	237806	124523	39088	23506	0.16436928	0.18876834
o4p25_15	203305	94541	50394	15469	0.24787388	0.16362213
o4p25_16	258713	149287	19893	32534	0.07689215	0.21792922
o4p25_17	317535	170871	37911	13738	0.11939156	0.08039983
o4p25_18	240983	132131	30729	26339	0.12751522	0.19934005
o4p25_19	262074	152592	34421	31191	0.13134077	0.20440783
o4p25_20	264813	129405	2234	825	0.00843614	0.00637533
o9p100_0	327678	132239	115127	41146	0.35134187	0.31114875
o9p100_1	305755	139085	90270	50320	0.29523638	0.36179315
o9p100_2	297880	138484	44734	12353	0.15017457	0.08920164

Tabla 14.- Análisis del frente no dominado de los algoritmo NSGA-II y NOSGA-II.

Como podemos observar en la Tabla 14, el algoritmo NSGA-II tiene un mayor porcentaje de soluciones en 9 instancias de 4 objetivos con 25 proyectos con un promedio de 0.15700726 y en las otras 11 instancias el algoritmo NOSGA-II obtuvo una mayor incidencia de soluciones con un promedio de 0.16096723.

Además tenemos que en las instancias de 9 objetivos con 100 proyectos el algoritmo que tuvo una mayor incidencia de soluciones en el frente cero general fue el algoritmo NOSGA-II, ya que en todas las instancias de 9 objetivos con 100 proyectos el algoritmo NOSGA-II supero a NSGA-II.

Continuando con el análisis del frente no dominado general se analizara en comportamiento de los algoritmos NS-ACO y NO-ACO, los resultados obtenidos se muestran en la Tabla 15.

Instancia	Soluciones generadas		Frente resultante		Porcentaje de Soluciones	
	ACO	NOACO	ACO	NOACO	ACO	NOACO
o4p25_0	44252	3266	262	139	0.00592064	0.04255971
o4p25_1	47920	2897	191	67	0.00398581	0.02312737
o4p25_2	45860	3361	257	208	0.00560401	0.06188634
o4p25_3	39803	3482	224	168	0.00562772	0.04824813
o4p25_4	31780	3141	139	57	0.00437382	0.01814709
o4p25_5	54823	3265	266	165	0.00485198	0.05053599
o4p25_7	35637	3201	159	36	0.00446166	0.01124649
o4p25_8	51221	3351	248	140	0.00484176	0.04177857
o4p25_9	28577	3190	152	65	0.00531896	0.02037618
o4p25_10	59230	3348	480	2	0.008104	0.00059737
o4p25_11	24295	3460	105	80	0.00432188	0.02312139
o4p25_12	24295	3460	105	80	0.00432188	0.02312139
o4p25_13	16577	4089	74	144	0.00446402	0.03521643
o4p25_14	53137	3222	329	87	0.00619154	0.02700186
o4p25_15	40118	3427	101	149	0.00251757	0.04347826
o4p25_16	43153	3201	195	14	0.00451881	0.00437363
o4p25_17	44370	3385	152	125	0.00342574	0.03692762
o4p25_18	36894	3265	191	119	0.00517699	0.03644717
o4p25_19	49100	3346	242	187	0.00492872	0.05588763
o4p25_20	42443	3231	173	11	0.00407605	0.00340452
o9p100_0	134093	3177	2450	915	0.0182709	0.28800755
o9p100_1	136137	3216	2608	925	0.01915717	0.28762438
o9p100_2	133666	3201	1803	677	0.01348885	0.21149641

Tabla 15.- Análisis del frente no dominado de los algoritmo NS-ACO y NO-ACO.

Como se puede observar en la Tabla 15 el algoritmo NS-ACO tuvo una incidencia mayor en solo dos instancias, las cuales son de 4 objetivos y 25 proyectos con una diferencia promedio de 0.00531079.

Además tenemos que en el resto de las instancias donde el algoritmo NO-ACO obtuvo una mayor incidencia de soluciones en el frente no dominado con una diferencia promedio de 0.06431123.

Por lo que podemos observar que el algoritmo que se vio más beneficiado en la incorporación de preferencias basado en el criterio del número de soluciones que permanecieron en el frente no dominado general fue el algoritmo NO-ACO, ya que supero en mayor medida al beneficio obtenido por el algoritmo NOSGA-II.

Para concluir con el análisis de los algoritmos secuenciales se procede a analizar la frontera no dominada estrictamente generada a partir del frente no dominado, este análisis iniciara con los algoritmos NOSGA-II y el NOACO, los resultados derivados de esta etapa de la experimentación e muestran en la Tabla 17.

Instancia	Soluciones Generadas		Porcentaje Repetidas		Soluciones NO repetidas		Soluciones en NS		Porcentaje Solu
	NOSGAI	NOACO	NOSGAI	NOACO	NOSGA	NOACO	NOSGAI	NOACO	NOSGAI
o4p25_0	45683	401	761.383333	10.8378378	60	37	0	4	0
o4p25_1	14239	258	569.56	13.5789474	25	19	0	5	0
o4p25_2	63189	465	1579.725	5.47058824	40	85	0	3	0
o4p25_3	96191	392	1354.80282	7.25925926	71	54	0	4	0
o4p25_4	33420	196	835.5	19.6	40	10	0	4	0
o4p25_5	79838	431	973.634146	4.31	82	100	0	3	0
o4p25_7	77834	195	1496.80769	17.7272727	52	11	0	3	0
o4p25_8	67444	388	1248.96296	6.25806452	54	62	0	3	0
o4p25_9	19712	217	469.333333	6.02777778	42	36	0	2	0
o4p25_10	91036	482	958.273684	241	95	2	1	1	1.05263158
o4p25_11	50769	185	2307.68182	20.5555556	22	9	0	3	1.052631580
o4p25_12	63519	266	1924.81818	10.64	33	25	2	2	6.06060606
o4p25_13	63519	218	1924.81818	6.22857143	33	35	0	1	0
o4p25_14	62594	416	978.03125	21.8947368	64	19	0	3	0
o4p25_15	65863	250	3874.29412	25	17	10	0	5	0
o4p25_16	52427	209	903.913793	26.125	58	8	1	1	1.72413793
o4p25_17	51649	277	1395.91892	5.32692308	37	52	0	3	0
o4p25_18	57068	310	2481.21739	14.0909091	23	22	0	3	0
o4p25_19	65612	52446	1151.08772	51.7218935	57	1014	0	6	0
o4p25_20	103059	184	624.6	30.6666667	165	6	0	2	0
o9p100_0	156273	1491	617.679842	1	253	1491	25	93	9.88142293
o9p100_1	140590	139353	399.403409	80.4578522	352	1732	27	140	7.67045455
o9p100_2	171342	136867	466.871935	104.31936	367	1312	27	140	7.35694823

Tabla 16.- Análisis de la Frontera no Superada Estrictamente.

Como se muestra en la Tabla 17 el algoritmo que tiene una incidencia mayor en el porcentaje de soluciones repetidas es el algoritmo NOSGA-II, también se observó que en la incidencia de soluciones no repetidas en algoritmo NOSGA-II tiene una incidencia mayor en 14 instancias de 4 objetivos con 25 proyectos con una incidencia promedio de 54.7857143.

Además tenemos que en las 6 instancias de 4 objetivos con 25 proyectos donde el algoritmo NO-ACO tuvo una mayor incidencia en el número de soluciones no repetidas, obtuvo una incidencia promedio de 224.666667.

También en la Tabla 17 se muestra que el algoritmo generaron soluciones con un mejor compromiso fue el algoritmo NO-ACO, ya que en solo tres instancias e encontró un empate con el algoritmo NOSGA-II debido a que la incidencia de numero de soluciones que permanecieron en la frontera no superada estrictamente fue la misma.

Por otro lado tenemos que la incidencia porcentual del algoritmo NO-ACO obtuvo una mayor incidencia en todas las instancias y esto se debe a que el número de soluciones generadas sin repetición fue menor en un 0.66666667 de las instancias resueltas.

Por lo que se puede concluir que el algoritmo que aproximó mejor la frontera no superada estrictamente en las experimentaciones realizadas fue el algoritmo NO-ACO, por lo que se propone como el algoritmo a paralelizar debido a que fue el algoritmo que presentó un mejor comportamiento no solo en términos del modelo de sobre-clasificación difusa (es decir, la frontera no superada estrictamente) si no que a su vez este algoritmo es el que se ve más beneficiado para aproximar mejor el frente no dominado.

7.3. Descripción de la aplicabilidad de los patrones de diseño al algoritmo NOACO

En esta sección se describirá la cada una de las implementaciones realizadas al algoritmo NO-ACO en su versión paralela, estas implementaciones se realizaron con la finalidad de reducir los tiempos de ejecución sin que el algoritmo no pierda la calidad en las soluciones generada.

Los patrones de diseños paralelos que se utilizaron para la implementación del algoritmo NO-ACO fueron los siguientes:

- Flujo de Datos.
- Grano Grueso.
- Maestro esclavo.

En total se realizaron 5 implementaciones paralelas para la meta heurística NO-ACO, dos de estas implementaciones se implementaron haciendo uso de la librería de OpenMP la cual hace uso de memoria compartida.

Por otro lado tenemos que otras dos partidas se implementaron haciendo el uso de la librería de MPI la cual hace uso de memoria distribuida, esta librería utiliza como protocolo de comunicación el paso de mensaje.

Y finalmente la implementación restante hace uso de las librerías de OpenMP y MPI generando una arquitectura híbrida entre el acceso a una memoria compartida y el intercambio de información mediante el paso de mensajes.

7.3.1. Implementación de los algoritmos basados en OpenMP.

El primer algoritmo implementado hace uso del patrón de diseño Grano Grueso (Acrónimo OMP-GG) el segmento del código que se paralelizó fue el proceso de la construcción de soluciones del *Algoritmo ACO*.

La aplicación de este algoritmo consistió en distribuir equitativamente la carga de trabajo en cada uno de los procesos disponibles en el entorno de ejecución, el trabajo delegado a cada uno de los procesos fue construir un segmento de la población.

Lo que quiere decir que cada procesador ejecuto $n = nH/nP$, donde n representa el número de soluciones, nH representa el número de hormigas de la colonia, nP representa el número de

procesos disponibles en el entorno de ejecución del algoritmo y n representa el número de bloques generados, una representación gráfica de la implementación se muestra en la Figura 17.

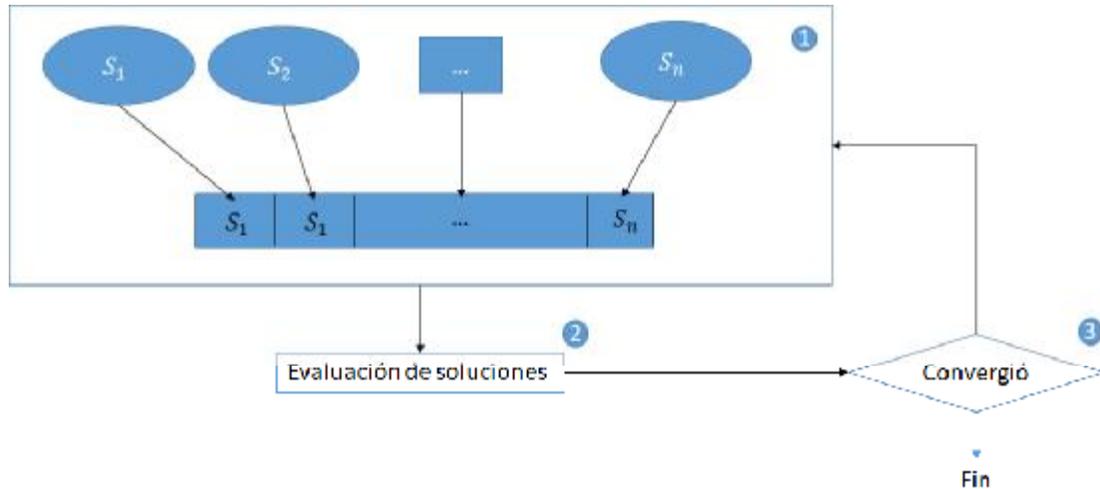


Figura 17.-Esquema de Grano Grueso.

En la Figura 17 se muestra el flujo de ejecución del patrón de diseño Grano Grueso de cada una de las iteraciones realizadas del algoritmo Meta-heurístico implementado, en el proceso 1 se muestra la división la división equitativa de la carga de trabajo, por lo que se generan subconjuntos de ejecución que son asignados a cada uno de los procesos de ejecución donde n hormigas construirán soluciones y cada una de estas compartirá de manera global la solución encontrada al resto de la colonia.

Posteriormente en el proceso 2 se procede a evaluar las soluciones generadas por las hormigas lo que le permite al algoritmo mejorar la calidad de las soluciones, este proceso se realiza de manera secuencial.

Finalmente en el proceso se evalúa el criterio de convergencia el cual es el encargado de detener el proceso evolutivo del algoritmo, el cual es evaluado al finalizar la evaluación de las soluciones. En esta implementación el criterio de convergencia está definido mediante el número de iteraciones, por lo que si este criterio se cumple finaliza el proceso de evolución del algoritmo en caso contrario se vuelve a ejecutar el proceso 1.

El segundo patrón de diseño utilizado para la implementación del otro algoritmo que hace uso de la librería de OpenMP, es el patrón de Flujo de Datos(Acrónimo OMP-FD) el cual tiene como finalidad asignar una *hormiga_i* a un proceso que se encuentre disponible en el entorno de ejecución este proceso se muestra en la Figura 18.

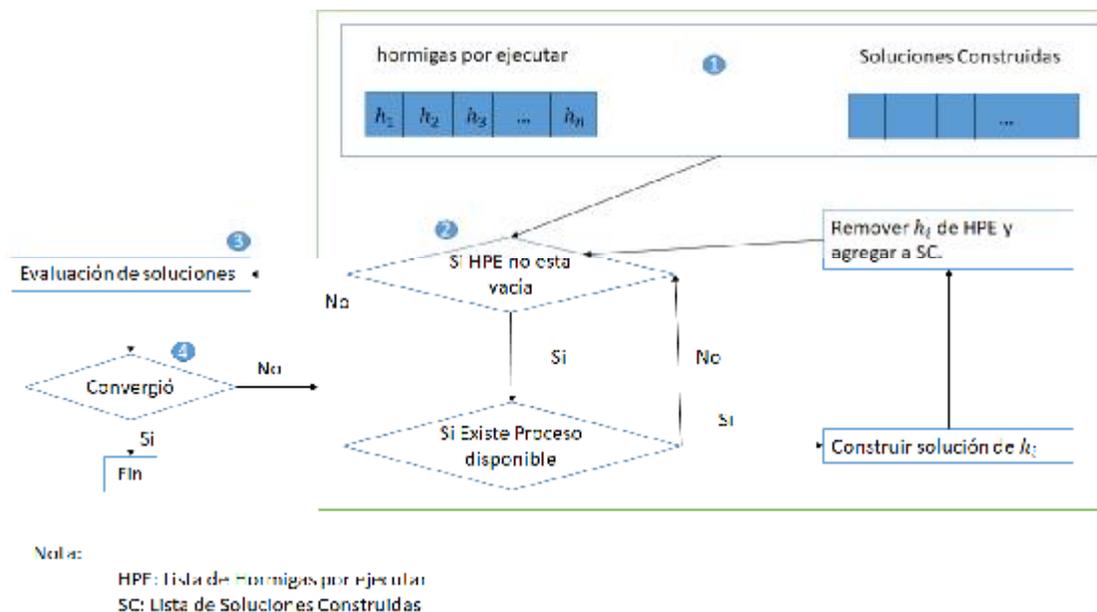


Figura 18.-Esquema Flujo de Datos.

En la Figura 18 se muestra el flujo de ejecución del patrón de diseño flujo de datos en cada una de las iteraciones realizadas del algoritmo Meta-heurístico implementado, este patrón de diseño tiene como finalidad asignar a un proceso que se encuentre disponible en el entorno de ejecución.

Por lo que cada hormiga tiene que esperar hasta que exista un proceso disponible para llevar a cabo la tarea de construcción de una solución, el proceso 1 tiene como finalidad inicializar las estructuras que representan el funcionamiento del algoritmo. Hormigas por ejecutar (HPE) se agrega toda la colonia de hormigas esperando ser solucionadas, por otro lado tenemos que en la estructura de Soluciones Construidas (SC).

Posteriormente en el proceso 2 inicia el proceso de construcción de soluciones para lo que primero se evalúa si existen hormigas que a un no han construido su solución, si todas las hormigas ya construyeron su soluciones es decir que la lista HPE se encuentra vacía entonces se procede al proceso 3 el cual ya fue descrito anteriormente.

Pero si aún existen hormigas en HPE se busca un procesador disponible para que la hormiga h_i pueda construir su solución, pero si todos los procesos se encuentran ocupados la hormiga h_i tiene que esperar hasta que un proceso se encuentre disponible.

Una vez que la hormiga h_i encontró un proceso disponible, esta hormiga procede a construir su solución. Por lo que esta hormiga es removida de la lista HPE y se agrega a la lista de *Soluciones Construidas* (SC), y posteriormente se verifica que aún existan hormigas por ejecutar. Por lo que de no existir hormigas en HPE se continúa con el proceso de evolutivo del algoritmo.

7.3.2. Implementación de los algoritmos basados en MPI.

El primer algoritmo basado haciendo uso de librería de MPI se hizo uso de la hibridación de los patrones de diseño Maestro Esclavo y Grano Grueso (Acrónimo MPI-MEGG), la finalidad de la aplicación del patrón de diseño del patrón de diseño Maestro Esclavo integrar las soluciones para evaluar la calidad de las soluciones.

La finalidad de aplicar el patrón de diseño de grano grueso es distribuir equitativamente la carga de trabajo en una población conjuntos de poblaciones de tamaño $S = nH/nP$, donde nH representa el número de hormigas de la colonia las cuales son las encargadas de construir las soluciones.

Una vez que una sub-colonia concluyo con la construcción de sus soluciones esta procede a notificar al nodo maestro que su actividad ya concluyo, este proceso se ilustra en la Figura 9.

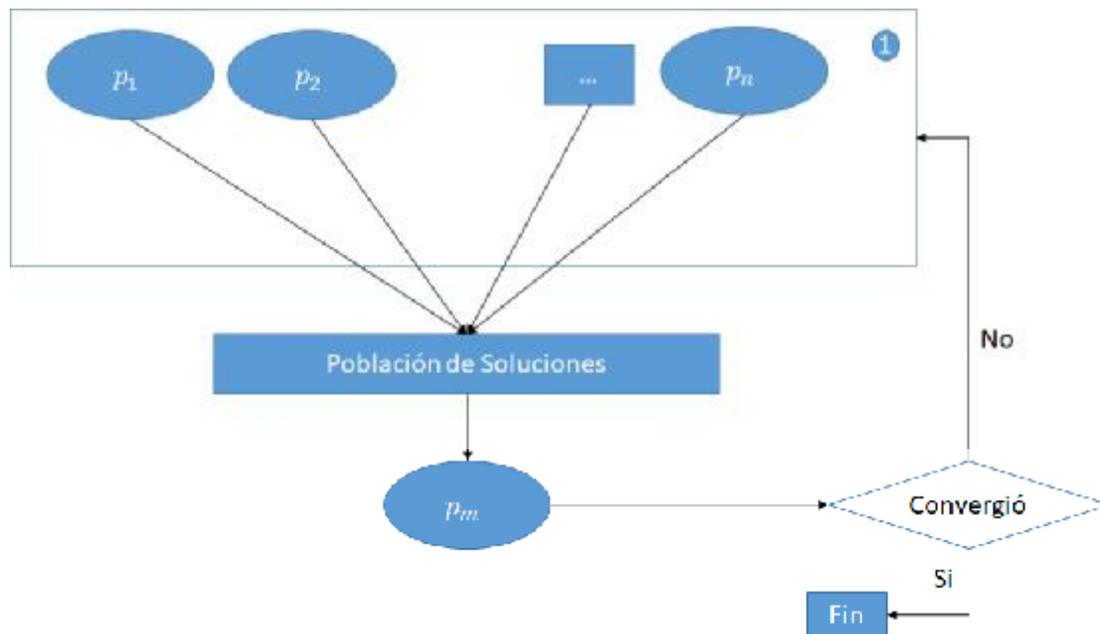


Figura 19. Esquema del Algoritmo MPI-MEGG.

El la Figura 19 se muestra que en la primera etapa uno del proceso se le asigna a cada uno de los procesadores disponibles en el entorno de ejecución una parte de la población de la colonia, cada una de estas sub-colonias se encuentran distribuidas de manera equitativa.

Una vez que una sub-colonia concluye con su actividad procede a notificar al nodo maestro que su actividad a concluido proporcionándole las soluciones que fueron construidas en su sub-colonia, una vez que todas las sub-colonias terminaron de construir y enviar las soluciones al nodo maestro P_m .

Este nodo procede a evaluar y actualizar la tabla de feromonas, una vez que el nodo maestro finalizo con el proceso de actualización de feromonas retroalimenta a todos los nodos esclavos con los

cambios efectuados en la tabla de feromonas, este proceso es realizado hasta alcanzar el criterio de convergencia.

El segundo algoritmo paralelo implementado hace uso de los patrones de diseño Flujo Datos y Maestro Esclavo (Acrónimo MPI-FDME), donde el patrón de diseño en el que recae la responsabilidad de la construcción de soluciones es el de Flujo de Datos.

Por otro lado tenemos que el patrón de diseño Maestro esclavo tiene como finalidad evaluar cada una de las evoluciones generadas por el algoritmo, este proceso se muestra en la Figura 20.

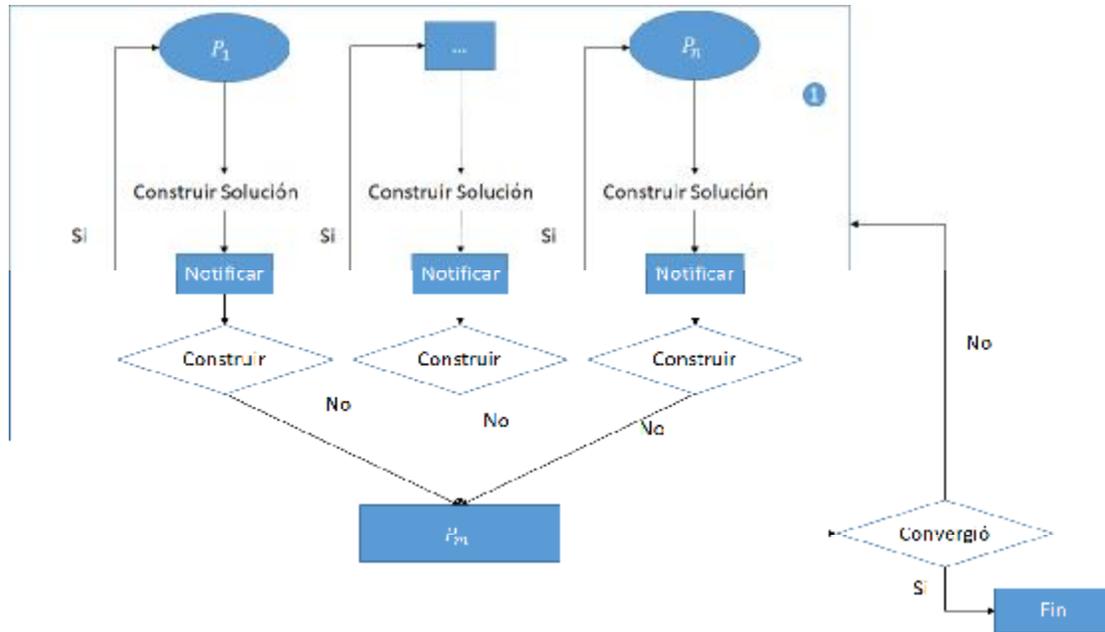


Figura 20.-Diagrama MPI-FDME.

En la Figura 20 el proceso 1 es el encargado de construir soluciones mediante la construcción de n islas de procesamiento las cuales se encargan de construir partes de la población de soluciones, este proceso ocurre en dos fases en cada uno de los nodos.

La primera fase consiste en la ejecución de una hormiga para que esta construya su solución una vez construida la solución se procede a notificar a P_m que se ha encontrado una solución, mediante un mensaje, el cual lleva como información la estructura de la solución encontrada.

En la segunda fase el nodo esclavo verifica si es necesario construir otra solución, por lo que si es necesaria la construcción se vuelve a repetir la primera fase. Por otro lado tenemos que si no es necesario construir otra solución este proceso se queda a la espera para volverse a ejecutar posteriormente.

Una vez que todos los nodos han terminado con la construcción de las soluciones el nodo maestro procede a evaluar las soluciones y actualizar la tabla de feromonas, además este nodo retroalimenta a cada uno de los nodos con la información de la actualización de la tabla de feromonas.

7.3.3. Implementación del Algoritmo Híbrido.

Para la implementación del último algoritmo paralelo se hizo uso de la arquitectura híbrida entre la librería de MPI y OMP en el cual se hizo uso de los patrones de diseño Grano Grueso, Flujo de Datos y Maestro Esclavo (Acrónimo H-MEGGFD).

La finalidad del patrón de diseño Grano Grueso es dividir equitativamente la carga de trabajo entre los procesadores disponibles en el entorno de ejecución, por otro lado tenemos que el patrón de diseño Flujo es realizar cada una de las construcciones que fueron asociadas al procesador.

Y finalmente el patrón de diseño Maestro Esclavo tiene como finalidad de hacer la integración de las soluciones, este proceso de comunicación entre los nodos se ilustra en la Figura 21.

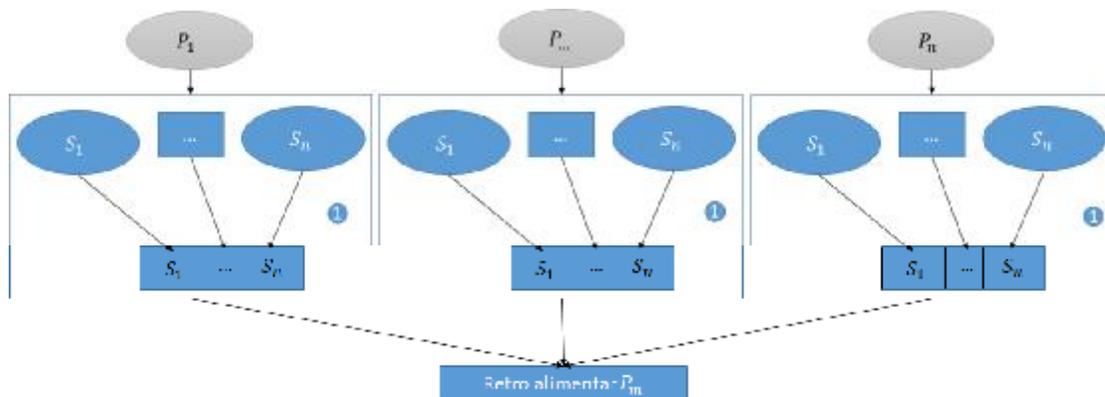


Figura 21.- Diagrama del Diseño H-MEGGFD

En la Figura 21 se muestra el proceso de comunicación que existe entre los nodos de procesamiento así como también el proceso de construcción de soluciones, este proceso inicia con el proceso de distribución de carga de trabajo entre los nodos de procesamiento P disponibles en el entorno de ejecución.

Posteriormente cada uno de los nodos empieza a construir sus soluciones haciendo uso de memoria compartida, una vez que el nodo término de construir sus soluciones comparte la información de las soluciones generadas con el nodo maestro.

Una vez que todos los nodos han concluido con la construcción de las soluciones, el nodo maestro puede empezar a evaluar todas las soluciones encontradas por los nodos esclavos, por lo que al finalizar este proceso el nodo maestro retroalimenta a los nodos esclavos con la información actualizada de la tabla de feromonas. Y este proceso se repite mientras no se cumpla el criterio de la convergencia.

Una vez implementados estos algoritmos se procedió a evaluar el desempeño de los algoritmos tomando como punto de comparación el algoritmo en su versión secuencial, ya que lo que se busca es encontrar el algoritmo que reduzca el tiempo de ejecución así como también que la calidad de las soluciones se vea afectada.

7.4. Análisis de los algoritmos Paralelos.

Esta actividad tiene como finalidad evaluar el rendimiento de los algoritmos paralelos implementados en términos de la calidad por lo que, el proceso utilizado para hacer este análisis se hizo de la misma manera que el análisis de los algoritmos secuenciales para evaluar el frente no dominado.

Además de evaluar la calidad de la solución también se tomaran en cuenta los tiempos de ejecución de los algoritmos implementados. Para hacer este análisis se hace uso del indicador del indicador aceleración. Para la realización de la prueba de hipótesis en esta experimentación hará uso de los siguientes indicadores:

- Aceleración máxima.
- Aceleración mínima.
- Aceleración promedio.
- Varianza de la aceleración.

Para la realización del análisis de los algoritmos paralelos se realizaron realización 31 experimentos, donde a cada uno de estos experimentos consta de 100 iteraciones para realizar el proceso evolutivo. Por otro lado tenemos que las instancias resueltas por los algoritmos fueron los mismos que fueron resueltas por los algoritmos secuenciales.

Para los algoritmos de MPI se realizaron experimentos con 2-N, 4-N, 6-N, 8-N, 16-N y 32-N (Donde N representa Nodo de procesamiento), por otro lado para los algoritmos basados en OMP se realizaron con 2-P, 4-P, 6-P, 8-P, 16-P y 32-P (Donde P representa un Proceso).

7.4.1. Análisis del Algoritmo OMP-FD.

Para iniciar el comportamiento del algoritmo OMP-FD se empezara el análisis de la calidad de las soluciones en términos del frente No Dominado, las cuales fueron generadas durante las experimentaciones realizadas con este patrón los resultados derivados de este análisis se muestran en la Tabla 18.

Instancia	S	2-P	4-P	8-P	16-P	32-P
o4p25_0	50%	38%	13%	0%	0%	0%
o4p25_1	22%	0%	33%	0%	0%	44%
o4p25_2	27%	27%	0%	36%	9%	0%
o4p25_3	20%	20%	0%	0%	20%	40%
o4p25_4	0%	0%	25%	0%	25%	50%
o4p25_5	33%	67%	0%	0%	0%	0%
o4p25_7	0%	50%	0%	0%	50%	0%

o4p25_8	0%	0%	33%	0%	0%	67%
o4p25_9	0%	100%	0%	0%	0%	0%
o4p25_10	0%	0%	33%	0%	67%	0%
o4p25_11	29%	14%	57%	0%	0%	0%
o4p25_12	100%	0%	0%	0%	0%	0%
o4p25_13	0%	0%	22%	0%	44%	33%
o4p25_14	0%	100%	0%	0%	0%	0%
o4p25_15	0%	100%	0%	0%	0%	0%
o4p25_16	17%	17%	33%	17%	17%	0%
o4p25_17	25%	0%	25%	0%	0%	50%
o4p25_18	38%	13%	25%	0%	13%	13%
o4p25_19	20%	40%	20%	0%	0%	20%
o4p25_20	0%	40%	20%	0%	20%	20%
o9p100_0	11%	89%	0%	0%	0%	0%
o9p100_1	6%	38%	44%	0%	6%	6%
o9p100_2	0%	0%	100%	0%	0%	0%

Tabla 17.- Incidencia de General del frente no dominado del Algoritmo OMP-GG.

Como podemos ver en la Tabla 17 el algoritmo secuencial implementado obtuvo una incidencia mayor en el frente no dominado en el 13% de las instancias que fueron resueltas, pero en todas las instancias que obtuvo la mayor incidencia fueron en las instancias de 4 objetivos con 25 proyectos.

Por otro lado tenemos que el algoritmo paralelo con el cual fue ejecutado con 2-P obtuvo un mejor comportamiento que el algoritmo secuencial ya que en el 33% de las instancias resueltas obtuvo una mayor incidencia de soluciones en el frente no dominado, donde una de las instancias resueltas fue de 9 objetivos y 100 proyectos.

Además tenemos que durante la ejecución del algoritmo paralelo con 4-P obtuvo un mejor comportamiento que el algoritmo secuencial en términos de la incidencia del frente no dominado, obteniendo en el 17% de las instancias resueltas una mayor incidencia.

También se observó que durante la ejecución del algoritmo con 8-P obtuvo un mal comportamiento en términos de la calidad de solución con parado con el algoritmo secuencial, ya que en la ejecución con 8-P se obtuvo que en un 4% de las instancias obtuvo una mayor incidencia de soluciones en el frente no dominado.

Pero por otro lado tenemos que durante la experimentación realizada con 16-P obtuvo un comportamiento similar que el algoritmo secuencial en términos de calidad de solución ya que también se observó que en un 13% de las soluciones obtuvo una mayor una incidencia de soluciones en el frente no dominado.

Y finalmente se observó que durante la experimentación realizada con 32-P el algoritmo paralelo obtuvo un mejor comportamiento en términos de la calidad de solución ya que durante esta experimentación se obtuvo que en el 21% de las instancias se obtuvo una mayor incidencia de soluciones en el frente no dominado. Ordenando las experimentaciones realizadas de mejor a peor experimentación los algoritmos quedan de la siguiente manera: I.-) 2-N; II.-) 32-N; III.-) Empate entre las experimentaciones IV) S y 16-N; 5.-) 8-P.

Continuando con el análisis del algoritmo OMP-FD se procederá con la comparación entre las experimentaciones realizadas en términos de tiempo de ejecución, haciendo uso del indicador de Aceleración obtenida la cual es calculada a partir del tiempo de ejecución del algoritmo secuencial. Este análisis iniciara con el indicador de aceleración mínima el cual se muestra en la Tabla 19.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.95833333	0.88461538	0.92	0.88461538	0.92
o4p25_1	0.92307692	0.92307692	0.92	0.92	0.92
o4p25_2	0.96	0.92307692	0.92307692	0.92307692	0.92307692
o4p25_3	0.92307692	0.96	0.92307692	0.92307692	0.92307692
o4p25_4	0.96	0.88888889	0.92307692	0.85714286	0.92307692
o4p25_5	0.92	0.92	0.92	0.92	0.95833333
o4p25_7	1	0.96	0.96	0.96	0.96
o4p25_8	0.92	0.92	0.92	0.92	0.92
o4p25_9	0.96	0.96	0.96	0.96	0.96
o4p25_10	0.96	0.92307692	0.96	0.92307692	0.96
o4p25_11	1	1	1	0.96296296	1
o4p25_12	0.96	1	0.92307692	0.96	0.96
o4p25_13	1	1	1	0.96153846	0.96153846
o4p25_14	0.96153846	0.96153846	0.96153846	0.96153846	0.96153846
o4p25_15	1	1	0.96	0.92307692	0.96
o4p25_16	0.92	0.92	0.92	0.95833333	0.95833333
o4p25_17	0.96	1	0.96	0.96	0.96
o4p25_18	1	0.96	0.96	0.92307692	0.96
o4p25_19	0.96	0.96153846	0.92307692	0.92307692	0.92307692
o4p25_20	0.96	0.96	0.96	0.96	0.96153846
o9p100_0	1.02912621	0.79699248	1.01941748	1.02941176	1.05
o9p100_1	1.03846154	1.02857143	1.01904762	1.02857143	1.05882353
o9p100_2	1.05309735	1.05309735	1.03539823	1.03539823	1.07272727

Tabla 18.- Indicador de Aceleración Mínima del algoritmo OMP-FD.

Como se muestra en la Tabla 19 tenemos que en todas las experimentaciones realizadas existieron algunas ejecuciones que tardaron más que la experimentación secuencial realizada, por lo que no alcanzaron una aceleración mínima de 1.

A continuación se procederá a evaluar el indicador de aceleración máxima que fue obtenido durante las experimentaciones realizadas, los resultados derivados en este indicador se muestran en la Tabla 20.

Instancia	2	4	8	16	32
o4p25_0	1.04166667	1.04347826	1.04166667	1.04166667	1
o4p25_1	1.04166667	1.04166667	1.08333333	1.04166667	1.08333333
o4p25_2	1.04166667	1.08333333	1.04	1.04166667	1.04166667
o4p25_3	1.04166667	1.04166667	1.04166667	1	1.04166667
o4p25_4	1.04166667	1.08333333	1.04166667	1.04166667	1.04166667
o4p25_5	1	1.04347826	1.04166667	1.04347826	1.04347826
o4p25_7	1.25	1.2	1.2	1.25	1.25
o4p25_8	1.04347826	1.04347826	1.04347826	1.04166667	1.04347826
o4p25_9	1.24	1.29166667	1.24	1.24	1.29166667
o4p25_10	1.125	1.08333333	1.125	1.08333333	1.08333333
o4p25_11	1.16	1.16	1.16666667	1.12	1.16
o4p25_12	1.29166667	1.25	1.25	1.29166667	1.25
o4p25_13	1.28	1.28	1.28	1.23076923	1.23076923
o4p25_14	1.2	1.25	1.2	1.2	1.2
o4p25_15	1.16666667	1.2173913	1.16666667	1.16666667	1.16666667
o4p25_16	1.30434783	1.25	1.25	1.30434783	1.30434783
o4p25_17	1.26086957	1.2173913	1.26086957	1.26086957	1.2173913
o4p25_18	1.26086957	1.20833333	1.20833333	1.16	1.20833333
o4p25_19	1.33333333	1.33333333	1.28	1.33333333	1.29166667
o4p25_20	1.375	1.375	1.375	1.375	1.33333333
o9p100_0	1.16666667	1.05309735	1.15533981	1.17821782	1.2020202
o9p100_1	1.14423077	1.14423077	1.14423077	1.14423077	1.17821782
o9p100_2	1.17272727	1.16216216	1.16216216	1.17272727	1.20560748

Tabla 19.- Indicador de Aceleración Máxima del algoritmo OMP-FD.

Como se puede observar en la Tabla 20 tenemos que la mejor aceleración obtenida en ese indicador fue de 1.29 y esta fue alcanzada por la experimentación realizada con 4-P, además podemos concluir que si obtuvo una aceleración con la paralización haciendo uso de este patrón de diseño.

Continuando con el análisis del algoritmo OMP-FD se procede a evaluar las experimentaciones realizadas haciendo uso del indicador Aceleración Promedio, los resultados obtenidos en este indicador se muestra en la Tabla 21.

Instancia	2	4	8	16	32
o4p25_0	0.99086022	0.98343313	0.97944582	0.96684864	0.97149711
o4p25_1	0.98864351	0.98595533	0.97842432	0.96720017	0.97356079
o4p25_2	0.98998346	1.00222544	0.97741935	0.98651365	0.9929239
o4p25_3	0.9914847	0.99913565	0.9801498	0.97980149	0.98791563
o4p25_4	0.98870141	0.98257789	0.97489247	0.96542065	0.97385856
o4p25_5	0.98182796	0.99787284	0.97671216	0.98086129	0.98312295
o4p25_7	1.06119107	1.06398784	1.05015715	1.04639371	1.05441328
o4p25_8	0.98339645	0.98839645	0.97796634	0.96880893	0.97398317
o4p25_9	1.0633871	1.07619682	1.05370968	1.05255859	1.06388499
o4p25_10	1.03096361	1.02702647	1.02047146	1.00738213	1.01596361

o4p25_11	1.07062862	1.06804797	1.0640488	1.04537451	1.05230769
o4p25_12	1.06677419	1.07333333	1.05435484	1.05446237	1.0594086
o4p25_13	1.09349462	1.08659222	1.08257191	1.06372484	1.07196765
o4p25_14	1.02934657	1.02692721	1.01885443	1.00830438	1.01352771
o4p25_15	1.0618864	1.055041	1.04935484	1.03742457	1.04282375
o4p25_16	1.02149135	1.03377407	1.00991585	1.01295933	1.01968911
o4p25_17	1.07789855	1.0787798	1.05895278	1.06063815	1.06606358
o4p25_18	1.05698223	1.04986209	1.04320009	1.03077085	1.03810659
o4p25_19	1.07978495	1.07429694	1.06858974	1.05641026	1.05794458
o4p25_20	1.13032258	1.12678246	1.11624069	1.10313069	1.11259053
o9p100_0	1.07083704	0.88931948	1.06543551	1.07730469	1.10617227
o9p100_1	1.07355727	1.07129339	1.07125729	1.07091121	1.0999633
o9p100_2	1.09190338	1.09602956	1.09129467	1.0923639	1.12140212

Tabla 20.- Indicador de Aceleración Promedio del algoritmo OMP-FD.

Como se puede observar en la Tabla 20 tenemos que la experimentación realizada con 2-P se obtuvo que en el 48% de las instancias resueltas obtuvo una mayor aceleración promedio, la experimentación realizada que obtuvo el segundo lugar en este indicador fue la realizada con 4-P.

Ya que la experimentación realizada con 4-P obtuvo que un 39% de las instancias resueltas obtuvo una mayor aceleración promedio a lo largo de las experimentaciones realizadas, además tenemos que en las experimentaciones realizadas con 2-P y 4-P salieron mejor renqueadas en las instancias de 4 objetivos.

Pero en las instancias de 9 objetivos con 100 la experimentación realizada con 32-P fue la que tuvo una mejor aceleración promedio ganándole a todas las experimentaciones en estas instancias, continuando con el análisis de este patrón se procede a evaluar la varianza generada en las experimentaciones. Los resultados derivados de este indicador se muestran en la Tabla 22.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.00062256	0.00111536	0.00091465	0.00120708	0.00053151
o4p25_1	0.00084977	0.00066948	0.00102985	0.00109829	0.00136117
o4p25_2	0.00071868	0.00140327	0.00075807	0.00125408	0.0009652
o4p25_3	0.00090283	0.00080029	0.00087272	0.00057366	0.00116773
o4p25_4	0.00086276	0.00116111	0.00094384	0.00140301	0.00113823
o4p25_5	0.00071082	0.00088568	0.00070276	0.00104798	0.00051629
o4p25_7	0.00299348	0.00280511	0.00307642	0.00380323	0.00332652
o4p25_8	0.00146823	0.00100309	0.00095314	0.00069828	0.00081278
o4p25_9	0.00390618	0.00421302	0.0040478	0.00390776	0.00500318
o4p25_10	0.0015224	0.00162794	0.00194666	0.00163862	0.00113589
o4p25_11	0.00152559	0.00178024	0.00176289	0.00138697	0.00239893
o4p25_12	0.00495644	0.00389803	0.00476726	0.00525545	0.00408612
o4p25_13	0.00385221	0.00387741	0.00400137	0.00394755	0.00422807
o4p25_14	0.00191146	0.0026054	0.00206899	0.00198586	0.0021168
o4p25_15	0.00244745	0.00248326	0.00244743	0.00297914	0.00282593

o4p25_16	0.00393283	0.00291463	0.0028684	0.0036158	0.00335881
o4p25_17	0.00559011	0.00441998	0.00479904	0.00522035	0.00469227
o4p25_18	0.00311801	0.00269225	0.00315153	0.00262598	0.00380695
o4p25_19	0.00616823	0.00578773	0.00556649	0.00616071	0.00648515
o4p25_20	0.00962122	0.01159074	0.01026364	0.00975435	0.00894042
o9p100_0	0.00117796	0.00339978	0.00111804	0.00119196	0.00129721
o9p100_1	0.00073893	0.00086893	0.00080086	0.0007717	0.00082876
o9p100_2	0.00101761	0.00106772	0.00103722	0.00131475	0.00111535

Tabla 21.- Indicador de la Varianza obtenida por la Aceleración del algoritmo OMP-FD.

Como se puede observar en la Tabla 22 tenemos que la varianza de la aceleración es muy pequeña comparada con las aceleraciones obtenidas, por lo que se puede decir que los tiempos de ejecución oscilaron en rangos pequeños.

Por los que al ordenar los algoritmos del mejor al peor en términos del tiempo de ejecución el orden queda de la siguiente manera: I) 2-P; II) 4-P; III) 32-P y finalmente hubo un empate en las experimentaciones realizadas con 8-P y 16-P.

7.4.2. Análisis del algoritmo OMP-GG

El análisis de las experimentaciones realizadas con este algoritmo se iniciara con el análisis en términos de la calidad de la solución y posteriormente se procederá con el análisis de los tiempos de ejecución haciendo uso del indicador de la aceleración alcanzada. Los resultados derivados en términos de la calidad de soluciones se muestran en la Tabla 23.

Instancia	S	2-N	4-N	8-N	16-N	32-N
o4p25_0	36%	27%	27%	9%	0%	0%
o4p25_1	22%	0%	33%	0%	0%	44%
o4p25_2	27%	27%	0%	36%	9%	0%
o4p25_3	20%	20%	0%	0%	20%	40%
o4p25_4	0%	0%	25%	0%	25%	50%
o4p25_5	33%	67%	0%	0%	0%	0%
o4p25_7	0%	33%	0%	33%	0%	33%
o4p25_8	0%	0%	33%	0%	0%	67%
o4p25_9	0%	100%	0%	0%	0%	0%
o4p25_10	0%	0%	33%	0%	67%	0%
o4p25_11	29%	14%	57%	0%	0%	0%
o4p25_12	100%	0%	0%	0%	0%	0%
o4p25_13	0%	0%	22%	0%	44%	33%
o4p25_14	0%	100%	0%	0%	0%	0%
o4p25_15	0%	100%	0%	0%	0%	0%
o4p25_16	17%	17%	33%	17%	17%	0%
o4p25_17	25%	0%	25%	0%	0%	50%
o4p25_18	38%	13%	25%	0%	13%	13%
o4p25_19	20%	40%	20%	0%	0%	20%

o4p25_20	0%	40%	20%	0%	20%	20%
o9p100_0	11%	89%	0%	0%	0%	0%
o9p100_1	6%	38%	44%	0%	6%	6%
o9p100_2	0%	0%	100%	0%	0%	0%

Tabla 22.- Frente No Dominado del algoritmo OMP-GG

Para iniciar con el análisis de la Tabla 23 tenemos que la incidencia de soluciones en el frente cero no dominado en las experimentaciones realizadas con el algoritmo secuencial obtuvo en el 12% de las soluciones obtenidas existió una mayor incidencia por este algoritmo.

Por otro lado tenemos que las experimentaciones realizadas con 2-P supero a las experimentaciones realizadas con el algoritmo secuencial, debido a que las experimentaciones realizadas con 2-P obtuvo que en el 32% de las instancias generadas obtuvo una mayor incidencia de soluciones en el frente cero no dominado.

Además tenemos que las experimentaciones realizadas con 4-P no supero la incidencia de soluciones obtenidas por el algoritmo secuencial, ya que las experimentaciones con 4-P solo obtuvieron que solo en el 16% de las instancias se obtuvo una mayor incidencia de soluciones en el frente no dominado.

También se puede observar que en las experimentaciones realizadas con 8-P y 16-P se obtuvieron que solo en el 8% de las instancias resueltas obtuvieron una incidencia de soluciones en el frente no dominado, pero estas experimentaciones no superaron a la incidencia de soluciones del algoritmo secuencial.

Finalmente se tiene que en las experimentaciones realizadas con 32-P se superó la incidencia de soluciones en el frente no dominado obtenido por las experimentaciones del algoritmo secuencial, ya que las experimentaciones realizadas con 32-P se obtuvieron que en el 24% de las instancias resueltas.

Por lo que el orden del comportamiento del mejor a la peor experimentación realizada que da de la siguiente manera: I) 2-P; II) 4-P; III); 32-P; IV) S; y finalmente V) existe un empate entre las experimentaciones realizadas con 8-P y 16-P.

Continuando con el análisis del algoritmo OMP-GG se continuara con el análisis de los tiempos de ejecución alcanzado en las experimentaciones realizadas haciendo uso del indicador aceleración, los resultados derivados de este análisis en el indicador Aceleración Mínima se muestran en la Tabla 24.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	1.15	1.17	1.27	0.85185185	0.92
o4p25_1	1.23	1.41	1.31	0.88461538	0.92307692
o4p25_2	1.24	1.37	1.22	0.88888889	0.92307692
o4p25_3	1.27	1.36	1.28	0.88888889	0.92307692
o4p25_4	1.22	1.31	1.3	0.88888889	0.92307692
o4p25_5	1.26	1.34	1.27	0.88461538	0.92307692

o4p25_7	1.21	1.34	1.21	0.96	0.96
o4p25_8	1.22	1.36	1.25	0.88461538	0.92
o4p25_9	1.23	1.37	1.35	0.92307692	0.96
o4p25_10	1.23	1.39	1.32	0.92307692	0.96
o4p25_11	1.29	1.26	1.33	0.96296296	1
o4p25_12	1.23	1.33	1.28	0.92307692	0.96
o4p25_13	1.2	1.39	1.3	0.92592593	1
o4p25_14	1.25	1.14	1.32	0.92592593	0.96153846
o4p25_15	1.25	1.32	1.26	0.92307692	0.96
o4p25_16	1.17	1.35	1.22	0.92	0.95833333
o4p25_17	1.25	1.36	1.27	0.92307692	1
o4p25_18	1.25	1.31	1.2	0.92307692	0.96
o4p25_19	1.2	1.39	1.27	0.96	1
o4p25_20	1.23	1.38	1.32	0.92592593	1
o9p100_0	1.68	2.33	2.28	0.90517241	0.9375
o9p100_1	1.75	2.24	2.3	0.99074074	1.01886792
o9p100_2	1.79	2.43	2.39	1.10377358	1.12380952

Tabla 23.-Indicador Aceleración Mínima el algoritmo OMP-GG.

Como se puede observar en la Tabla 23 que en las experimentaciones realizadas con 2-P, 4-P y 8-P obtuvieron una aceleración mayor en este indicador por lo que, en todas las experimentaciones realizadas con estos procesadores existió una aceleración respecto al algoritmo secuencial.

Por otro lado tenemos que en las experimentaciones realizadas con 16-P y 32-P existió una desaceleración en términos del tiempo de ejecución, por lo que se dice que en estas ejecuciones en algunas experimentaciones fue más lento que el algoritmo secuencial.

Continuando con este análisis se procede a evaluar el indicador de incidencia máxima del algoritmo OMP-GG, los resultados obtenidos se muestran en la Tabla 25.

Instancia	2	4	8	16	32
o4p25_0	1.49	1.65	1.49	1	1.04166667
o4p25_1	1.5	1.67	1.57	1	1.04166667
o4p25_2	1.5	1.63	1.55	1.04	1.08333333
o4p25_3	1.5	1.6	1.6	1	1.04166667
o4p25_4	1.51	1.64	1.56	1	1.04166667
o4p25_5	1.58	1.72	1.54	1	1.04166667
o4p25_7	1.78	1.97	1.76	1.2	1.25
o4p25_8	1.43	1.6	1.54	1	1.08695652
o4p25_9	1.52	1.7	1.59	1.19230769	1.25
o4p25_10	1.49	1.71	1.59	1.04	1.125
o4p25_11	1.61	1.85	1.68	1.08	1.16
o4p25_12	1.48	1.65	1.64	1.19230769	1.29166667
o4p25_13	1.57	1.78	1.65	1.23076923	1.25

o4p25_14	1.51	1.63	1.66	1.15384615	1.2
o4p25_15	1.57	1.79	1.79	1.125	1.17391304
o4p25_16	1.59	1.71	1.57	1.25	1.30434783
o4p25_17	1.61	1.72	1.55	1.16	1.2173913
o4p25_18	1.63	1.76	1.62	1.16	1.20833333
o4p25_19	1.51	1.76	1.51	1.28	1.33333333
o4p25_20	1.77	1.89	1.7	1.32	1.375
o9p100_0	1.8	2.47	2.44	1.04385965	1.0625
o9p100_1	1.97	2.67	2.61	1.11214953	1.14423077
o9p100_2	1.91	2.68	2.58	1.21904762	1.26470588

Tabla 24.- Indicador Aceleración Máxima el algoritmo OMP-GG.

Como se puede observar en la Tabla 25 en la experimentación realizada con 2-P, 4-P y 8-P obtuvo que en todas las instancias resueltas alcanzo, mejorando así el tiempo de ejecución en al menos en una de las experimentaciones realizadas.

Por otro lado tenemos que en las experimentaciones realizadas con 16-P en estas se obtuvo una perdida en términos de la aceleración ya que en el 25% de las instancias resueltas a lo más se alcanzó igualar el tiempo de ejecución por lo que no se percibió la aceleración.

Por otro lado tenemos que en las experimentaciones realizadas con 32-P se alcanzó obtener en al menos en una de las experimentaciones realizadas, ya que en este indicador en todas las instancias supero a 1.

Además se puede observar que las mejores aceleraciones obtenidas en este indicador son alcanzadas por las experimentaciones con 4-P y 8-P en las instancias de 9 objetivos y 100 proyectos ya que se alcanzó a superar la aceleración de 2.

A continuación se presenta el análisis del indicador de aceleración promedio derivado de las experimentaciones realizadas, los resultados de este indicador se muestran en la Tabla 26.

Instancia	2	4	8	16	32
o4p25_0	1.34	1.47	1.38	0.9423178	0.97816377
o4p25_1	1.36	1.51	1.45	0.94390957	0.99122829
o4p25_2	1.36	1.49	1.39	0.96424226	0.9902895
o4p25_3	1.37	1.5	1.42	0.94972153	0.98864351
o4p25_4	1.36	1.52	1.42	0.94324603	0.9887469
o4p25_5	1.39	1.5	1.39	0.95061621	0.98578991
o4p25_7	1.36	1.54	1.42	1.02074442	1.06123656
o4p25_8	1.31	1.46	1.36	0.9503598	0.99006887
o4p25_9	1.37	1.52	1.45	1.02407775	1.06370554
o4p25_10	1.36	1.53	1.43	0.98163036	1.02961952
o4p25_11	1.41	1.56	1.46	1.02487088	1.07360629
o4p25_12	1.33	1.48	1.43	1.02173697	1.06510753
o4p25_13	1.35	1.58	1.45	1.03911038	1.09224979
o4p25_14	1.38	1.48	1.43	0.98520357	1.02784533

o4p25_15	1.41	1.54	1.44	1.00686104	1.0595582
o4p25_16	1.33	1.48	1.36	0.97972291	1.02126695
o4p25_17	1.35	1.48	1.42	1.03112903	1.07564282
o4p25_18	1.38	1.48	1.41	1.00727461	1.0541094
o4p25_19	1.33	1.51	1.38	1.02803603	1.07828784
o4p25_20	1.4	1.53	1.45	1.0766391	1.1272043
o9p100_0	1.74	2.41	2.35	0.95953629	0.97610742
o9p100_1	1.81	2.48	2.42	1.03683821	1.06399338
o9p100_2	1.84	2.55	2.49	1.15389693	1.18283695

Tabla 25.- Indicador de Aceleración Promedio del algoritmo OMP-GG.

Como se puede observar en la Tabla 25 las experimentaciones realizadas con 4-P obtuvo un mejor desempeño ya que en el 100% de las instancias resueltas supero al resto de las experimentaciones lo cual da evidencia de que es el algoritmo con mejor desempeño en tiempo de ejecución.

Por otro lado tenemos que las experimentaciones realizadas con 8-P no pudo superar las experimentaciones realizadas con 4-P, pero las experimentaciones realizadas con 8-P supero en el 100% de las instancias resueltas al resto de las soluciones.

Además se tiene que en las experimentaciones realizadas con 2-P supero a las experimentaciones con 16-P y 32-P superando a estas en todas las instancias que fueron resueltas, también se puede observar que en las experimentaciones realizadas con 32-P esta alcanzo una aceleración promedio en el 69.56% de las instancias resueltas.

Finalmente se puede concluir que el algoritmo que tuvo el peor comportamiento fueron las experimentaciones realizadas con 16-P ya que en solo el 52.17% de las instancias resueltas se pudo alcanzar una aceleración promedio.

Continuando con el análisis de los indicadores del algoritmo OMP-GG, se analizara el indicador de la varianza de la aceleración, estos resultados se muestran en la Tabla 27.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.008	0.01	0.003	0.00108528	0.00102194
o4p25_1	0.006	0.005	0.005	0.00081447	0.00091008
o4p25_2	0.004	0.004	0.004	0.00111944	0.00112403
o4p25_3	0.002	0.003	0.003	0.00092349	0.00063454
o4p25_4	0.004	0.005	0.004	0.00053796	0.00084382
o4p25_5	0.006	0.006	0.004	0.0009478	0.00067069
o4p25_7	0.01	0.015	0.014	0.00315921	0.00321433
o4p25_8	0.008	0.006	0.007	0.00095845	0.00142726
o4p25_9	0.005	0.007	0.004	0.00309463	0.00452284
o4p25_10	0.004	0.006	0.004	0.00124288	0.00156539
o4p25_11	0.006	0.011	0.007	0.00154048	0.00207965
o4p25_12	0.003	0.006	0.007	0.00349788	0.00436916
o4p25_13	0.006	0.008	0.006	0.0045537	0.00372694
o4p25_14	0.003	0.008	0.004	0.00138964	0.00172119

o4p25_15	0.007	0.01	0.01	0.002615	0.00316163
o4p25_16	0.008	0.006	0.007	0.00325418	0.00372602
o4p25_17	0.005	0.006	0.004	0.0036348	0.00413225
o4p25_18	0.008	0.009	0.009	0.00339404	0.00293694
o4p25_19	0.005	0.005	0.003	0.00480425	0.00568007
o4p25_20	0.009	0.009	0.005	0.00930705	0.00920545
o9p100_0	0.001	0.001	0.001	0.000883	0.00089437
o9p100_1	0.002	0.008	0.005	0.00088377	0.00084336
o9p100_2	0.001	0.003	0.002	0.00116539	0.00133221

Tabla 26.- Indicador de la varianza de la Aceleración del algoritmo OMP-GG.

Como podemos observar en la Tabla 26 las varianzas obtenidas son muy pequeñas por los que el ranking de las experimentaciones realizadas en tiempos de ejecución queda de la siguiente manera: I) 4-P; II) 8-P; III) 2-P IV) 32-P; y finalmente V) 16-P.

7.4.3. Análisis del Algoritmo MPI-MEGG.

Para iniciar con este análisis de este algoritmo se procede a evaluar la calidad de solución en términos de la incidencia de soluciones no dominadas en el frente no dominado general, pero se observó que solo las que solo las soluciones que permanecieron en este frente fueron las soluciones generadas por el algoritmo secuencial estos resultados se muestra en la Tabla 28.

Instancia	Iniciales	Secuencial	Porcentaje
o4p25_0	5989	8	0.134%
o4p25_1	4195	4	0.095%
o4p25_2	5335	2	0.037%
o4p25_3	6971	5	0.072%
o4p25_4	4128	1	0.024%
o4p25_5	5939	3	0.051%
o4p25_7	5897	1	0.017%
o4p25_8	6578	3	0.046%
o4p25_9	5631	3	0.053%
o4p25_10	4292	2	0.047%
o4p25_11	4298	2	0.047%
o4p25_12	6889	5	0.073%
o4p25_13	3741	2	0.053%
o4p25_14	4867	3	0.062%
o4p25_15	5330	1	0.019%
o4p25_16	6648	3	0.045%
o4p25_17	5795	5	0.086%
o4p25_18	5591	8	0.143%
o4p25_19	6184	6	0.097%
o4p25_20	6092	4	0.066%
o9p100_0	16057	9	0.056%

o9p100_1	16366	4	0.024%
o9p100_2	5377	1	0.019%

Tabla 27.-Frente No Dominado del algoritmo MPI-MEGG.

En la Tabla 28 se muestra el número de soluciones que se generaron en el frente no dominado durante la experimentación del algoritmo secuencial el cual esta representada en la Tabla como **Inicial**, además también se muestra el tamaño del frente no dominado general como el indicador **Final**.

El indicador final se obtuvo generado el frente no dominado general el cual se generó a través de las soluciones generadas por el algoritmo secuencial y las experimentaciones del algoritmo paralelo MPI-MEGG realizadas con 2-N, 4-N, 8-N, 16-N, y 32-N.

Como se puede observar en la Tabla 27 el número de soluciones permanentes en el frente no dominado oscila entre 1 y 9 soluciones, por lo que la varianza obtenida del el número de soluciones es 5.40316206 soluciones.

Además se puede observar que el porcentaje de incidencia de soluciones en el frente no dominado (porcentaje) es muy bajo ya que en solo una de las instancias solamente se obtuvo una incidencia mayor al 1% de las soluciones generadas.

Por lo que se procedió a hacer un nuevo análisis de la calidad de la solución en la cual el algoritmo secuencial fue omitido para la construcción del frente no dominado general con la finalidad de obtener un ranking haciendo el cual se establece mediante la incidencia del nuevo frente generado, los resultados obtenidos se muestran en la Tabla 29.

Instancia	2-N	4-N	8-N	16-N	32-N
o4p25_0	15%	14%	28%	20%	23%
o4p25_1	84%	0%	8%	2%	6%
o4p25_2	67%	9%	8%	11%	5%
o4p25_3	16%	20%	26%	20%	18%
o4p25_4	24%	19%	21%	19%	17%
o4p25_5	0%	35%	19%	23%	24%
o4p25_7	90%	1%	0%	9%	0%
o4p25_8	14%	19%	22%	17%	28%
o4p25_9	100%	0%	0%	0%	0%
o4p25_10	0%	22%	30%	26%	23%
o4p25_11	14%	25%	27%	17%	17%
o4p25_12	45%	10%	23%	10%	11%
o4p25_13	49%	10%	17%	15%	9%
o4p25_14	13%	24%	15%	26%	22%
o4p25_15	74%	5%	10%	5%	5%
o4p25_16	0%	25%	21%	32%	23%
o4p25_17	82%	4%	8%	3%	4%
o4p25_18	10%	21%	24%	23%	23%
o4p25_19	86%	4%	4%	3%	3%

o4p25_20	5%	22%	24%	26%	22%
o9p100_0	100%	0%	0%	0%	0%
o9p100_1	0%	0%	100%	0%	0%
o9p100_2	2%	39%	39%	0%	20%

Tabla 28.- Incidencia de soluciones del Frente No Dominado del algoritmo MPI-MEGG.

Como se puede observar en la Tabla 28, el algoritmo que tuvo una incidencia mayor en las experimentaciones realizadas fue la experimentación realizada con 2-N, ya que obtuvo una incidencia mayor en el 46% de las instancias resueltas durante las experimentaciones realizadas,

Además se observa que en las experimentaciones realizadas con 8-N se observó que en el 29% de las instancias resueltas obtuvo una mayor incidencia de soluciones en el frente no dominado, por otro lado se observó que en las experimentaciones realizadas con 16-N se obtuvo que en el 13% de las instancias resueltas se alcanzó una mayor incidencia de soluciones en el frente no dominado.

También se observó que en las experimentaciones realizadas con 4-N solo se alcanzó a obtener una mayor incidencia en el frente no dominado general en el 8% de las instancias resueltas. Finalmente tenemos que durante las experimentaciones realizadas con 32-N se obtuvo que en el solo 4% de las instancias resueltas se una mayor incidencia de soluciones en este frente.

Por lo que el ranking de los algoritmos haciendo bajo el criterio de soluciones de la incidencia de soluciones en el frente cero no dominado general queda de la siguiente manera: I) Secuencial; II) 2-N; III) 8-N; IV) 16-N; V) 4-N; y finalmente VI) 32-N.

Una vez realizado el análisis de la calidad de las soluciones generadas se procede a realizar los análisis correspondientes a los tiempos de ejecución generados durante las experimentaciones realizadas al algoritmo MPI-MEGG, dichos resultados que se derivan del indicador aceleración minina se muestran en la Tabla 30.

Instancia	2	4	8	16	32
o4p25_0	1.49	1.65	1.27	1.33333333	1
o4p25_1	1.5	1.67	1.31	1.18181818	1
o4p25_2	1.5	1.63	1.22	1.04	1.04
o4p25_3	1.5	1.6	1.28	0.86206897	1
o4p25_4	1.51	1.64	1.3	0.78125	1
o4p25_5	1.58	1.72	1.27	0.66666667	1
o4p25_7	1.78	1.97	1.21	0.775	1.19230769
o4p25_8	1.43	1.6	1.25	0.71428571	1.2
o4p25_9	1.52	1.7	1.35	0.53191489	1
o4p25_10	1.49	1.71	1.32	0.6	1.19230769
o4p25_11	1.61	1.85	1.33	0.48214286	1.04
o4p25_12	1.48	1.65	1.28	0.49152542	1.08
o4p25_13	1.57	1.78	1.3	0.50819672	1.19230769
o4p25_14	1.51	1.63	1.32	0.49230769	1.23076923
o4p25_15	1.57	1.79	1.26	0.44117647	1.15384615
o4p25_16	1.59	1.71	1.22	0.3943662	1.125
o4p25_17	1.61	1.72	1.27	0.39473684	1.25

o4p25_18	1.63	1.76	1.2	0.41428571	1.16
o4p25_19	1.51	1.76	1.27	0.42028986	1.16
o4p25_20	1.77	1.89	1.32	0.43835616	1.28
o9p100_0	1.8	2.47	2.28	0.42105263	1.32
o9p100_1	1.97	2.67	2.3	0.74375	1.04385965
o9p100_2	1.91	2.68	2.39	1.11214953	1.11214953

Tabla 29.- Indicador Aceleración Mínima el algoritmo MPI-MEG.

Como se puede observar en la Tabla 29 en las experimentaciones realizadas al algoritmo MPI-GG se alcanzó una aceleración en las experimentaciones con 2-N, 4-N y 8-N, lo que indica que en estas experimentaciones el algoritmo tardó menos tiempo de ejecución.

Además se observó que en las experimentaciones de 2-N, 4-N y 8-N las instancias que obtuvieron una aceleración en este indicador fueron las instancias de 9 objetivos con 100 proyectos. También se observó que en las experimentaciones realizadas con 16-P y 32-N que en al menos en una de estas experimentaciones el algoritmo tardó más tiempo de ejecución comparado con el algoritmo secuencial. Continuando con el análisis del algoritmo MPI-MEGG se procede con el análisis del indicador aceleración máxima, los resultados derivados de este indicador se muestran en la Tabla 31.

Instancia	2-P	4-P	8	16-P	32-P
o4p25_0	3.37	3.12	1.49	1.04545455	0.85185185
o4p25_1	3.47	2.75	1.57	0.92	0.88461538
o4p25_2	3.32	3.02	1.55	0.88888889	0.88888889
o4p25_3	3.35	2.93	1.6	0.75757576	0.88888889
o4p25_4	4.47	3.19	1.56	0.64864865	0.88888889
o4p25_5	2.46	2.76	1.54	0.575	0.88461538
o4p25_7	3.95	3.02	1.76	0.5952381	0.96153846
o4p25_8	3.15	2.56	1.54	0.52173913	0.96
o4p25_9	3.44	3.25	1.59	0.46938776	0.88461538
o4p25_10	3.55	2.46	1.59	0.45283019	0.92307692
o4p25_11	5.66	3.62	1.68	0.42105263	0.92307692
o4p25_12	3.79	3	1.64	0.41935484	0.96296296
o4p25_13	7.33	3.61	1.65	0.36923077	0.92307692
o4p25_14	3.43	3	1.66	0.37878788	0.92592593
o4p25_15	3.65	3.28	1.79	0.34722222	0.92592593
o4p25_16	3.29	2.48	1.57	0.32432432	0.92307692
o4p25_17	3.66	2.91	1.55	0.30263158	0.92
o4p25_18	3.17	2.73	1.62	0.32432432	0.92307692
o4p25_19	4.19	3	1.51	0.32876712	0.92307692
o4p25_20	3.85	2.73	1.7	0.32432432	0.96
o9p100_0	8.37	6.19	2.44	0.31168831	0.92592593
o9p100_1	8.15	6.97	2.61	0.53535354	0.90517241
o9p100_2	9.23	7.58	2.58	0.99074074	0.99074074

Tabla 30.- Indicador Aceleración Máxima el algoritmo MPI-MEGG.

En base a la Tabla 31 se puede concluir que las experimentaciones realizadas con 2-N, 4-N y 8-N al algoritmo MPI-MEGG obtuvieron mejores resultados en este indicador ya que en estas experimentaciones se disminuyeron los tiempos de ejecución comparada con el algoritmo secuencia.

Además también se observa que en esas experimentaciones las instancias en las que se disminuyó considerablemente el tiempo de ejecución fueron en las instancias de 9 objetivos con 100 proyectos.

Siguiendo con el análisis de los tiempos de ejecución alcanzados por el algoritmo MPI-MEGG durante las experimentaciones realizadas, se procede con el análisis del indicador de aceleración promedio, los resultados obtenidos del análisis se muestran en la Tabla 32.

Instancia	2	4	8	16	32
o4p25_0	1.34	1.47	1.38	1.21238813	0.9423178
o4p25_1	1.36	1.51	1.45	1.01503724	0.94390957
o4p25_2	1.36	1.49	1.39	0.96424226	0.96424226
o4p25_3	1.37	1.5	1.42	0.80968744	0.94972153
o4p25_4	1.36	1.52	1.42	0.70177828	0.94324603
o4p25_5	1.39	1.5	1.39	0.62982159	0.95061621
o4p25_7	1.36	1.54	1.42	0.63871233	1.03548387
o4p25_8	1.31	1.46	1.36	0.58361474	1.02074442
o4p25_9	1.37	1.52	1.45	0.49926803	0.9503598
o4p25_10	1.36	1.53	1.43	0.5060869	1.02407775
o4p25_11	1.41	1.56	1.46	0.44961533	0.98163036
o4p25_12	1.33	1.48	1.43	0.45573961	1.02487088
o4p25_13	1.35	1.58	1.45	0.41619479	1.02173697
o4p25_14	1.38	1.48	1.43	0.410182	1.03911038
o4p25_15	1.41	1.54	1.44	0.36371762	0.98520357
o4p25_16	1.33	1.48	1.36	0.34885661	1.00686104
o4p25_17	1.35	1.48	1.42	0.31905724	0.97972291
o4p25_18	1.38	1.48	1.41	0.36031206	1.03112903
o4p25_19	1.33	1.51	1.38	0.35725516	1.00727461
o4p25_20	1.4	1.53	1.45	0.35624575	1.02803603
o9p100_0	1.74	2.41	2.35	0.34708032	1.0766391
o9p100_1	1.81	2.48	2.42	0.62029819	0.95953629
o9p100_2	1.84	2.55	2.49	1.03683821	1.03683821

Tabla 31.-Indicador de Aceleración del algoritmo MPI-MEGG.

Como se puede observar la Tabla 32 la experimentación realizada con 4-N fue la que tuvo un mejor comportamiento en el 100% de las instancias soluciones, también se puede observar que la experimentación realizada con 8-N supero en el 100% de las instancias al resto de las experimentaciones realizadas.

Por otro lado tenemos que en la experimentación realizada con 2-N supero considerablemente a las experimentaciones realizadas con 16-N y 32-N ya que estas en estas dos experimentaciones no se mejoraron en promedio el tiempo de ejecución. Otra de las observaciones a la experimentación

realizada con 2-N se observó que la aceleración disminuyó de manera considerable la aceleración en las instancias de 9 objetivos y 100 proyectos.

Continuando con el análisis de la aceleración obtenida en las experimentaciones obtenidas, se continuara con el análisis del indicador varianza. Los resultados obtenidos en este indicador se muestran en la Tabla 33.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.008	0.01	0.003	0.00521326	0.00108528
o4p25_1	0.006	0.005	0.005	0.0037348	0.00081447
o4p25_2	0.004	0.004	0.004	0.00111944	0.00111944
o4p25_3	0.002	0.003	0.003	0.00104071	0.00092349
o4p25_4	0.004	0.005	0.004	0.00065624	0.00053796
o4p25_5	0.006	0.006	0.004	0.00046879	0.0009478
o4p25_7	0.01	0.015	0.014	0.00119089	0.00247268
o4p25_8	0.008	0.006	0.007	0.0014703	0.00315921
o4p25_9	0.005	0.007	0.004	0.00027323	0.00095845
o4p25_10	0.004	0.006	0.004	0.0009055	0.00309463
o4p25_11	0.006	0.011	0.007	0.00028407	0.00124288
o4p25_12	0.003	0.006	0.007	0.00031198	0.00154048
o4p25_13	0.006	0.008	0.006	0.00071028	0.00349788
o4p25_14	0.003	0.008	0.004	0.00052686	0.0045537
o4p25_15	0.007	0.01	0.01	0.00026779	0.00138964
o4p25_16	0.008	0.006	0.007	0.0002862	0.002615
o4p25_17	0.005	0.006	0.004	0.00027251	0.00325418
o4p25_18	0.008	0.009	0.009	0.00039103	0.0036348
o4p25_19	0.005	0.005	0.003	0.00041065	0.00339404
o4p25_20	0.009	0.009	0.005	0.00059642	0.00480425
o9p100_0	0.001	0.001	0.001	0.00084191	0.00930705
o9p100_1	0.002	0.008	0.005	0.00225539	0.000883
o9p100_2	0.001	0.003	0.002	0.00088377	0.00088377

Tabla 32.- Indicador de la varianza obtenida de la Aceleración del algoritmo MPI-MEGG.

Como se puede observar en la Tabla 33 en este indicador se obtuvo que la aceleración no vario considerablemente en los tiempos de ejecución lo que indica que los tiempos de ejecución no variaron significativamente.

Por lo que el ranquin de las experimentaciones bajo el criterio del indicador de la aceleración queda de la siguiente manera: I) 4-N; II) 8-N; III) 2-N; y finalmente existió un empate entre las experimentaciones realizadas con IV) 16-N y 32-N.

7.4.4. Análisis del Algoritmo MPI-MEFD.

Para realizar el análisis del algoritmo MPI-MEFD se iniciara con el análisis en términos de la calidad de la solución mediante el análisis del frente no dominado general, en los resultados obtenidos de las experimentaciones realizadas se observó que en todas las experimentaciones realizadas solamente inciden las soluciones generadas por el algoritmo secuencial en el frente no dominado general.

Por lo que se procedió a evaluar el frente no dominado general excluyendo a las soluciones generadas por el algoritmo secuencial. Por lo que se procedió a generar un nuevo frente no dominado general, los resultados obtenidos de este nuevo frente se muestran en la Tabla 34.

Instancia	2-N	4-N	8-N	16-N	32-N
o4p25_0	100%	0%	0%	0%	0%
o4p25_1	68%	6%	13%	4%	9%
o4p25_2	52%	17%	8%	10%	14%
o4p25_3	44%	10%	12%	17%	17%
o4p25_4	11%	28%	11%	19%	30%
o4p25_5	0%	25%	25%	4%	45%
o4p25_7	72%	8%	11%	3%	6%
o4p25_8	1%	22%	19%	29%	30%
o4p25_9	100%	0%	0%	0%	0%
o4p25_10	0%	35%	31%	19%	16%
o4p25_11	11%	19%	22%	19%	29%
o4p25_12	78%	3%	2%	11%	6%
o4p25_13	43%	22%	8%	15%	13%
o4p25_14	15%	27%	7%	25%	25%
o4p25_15	73%	6%	9%	10%	3%
o4p25_16	1%	20%	25%	26%	28%
o4p25_17	75%	9%	2%	8%	6%
o4p25_18	27%	17%	18%	19%	20%
o4p25_19	59%	10%	13%	11%	7%
o4p25_20	16%	25%	22%	17%	21%
o9p100_0	100%	0%	0%	0%	0%
o9p100_1	40%	13%	14%	18%	16%
o9p100_2	64%	9%	9%	8%	10%

Tabla 33.- Frente No Dominado del algoritmo MPI-MEFD.

Como se puede observar en la Tabla 34 la experimentación que obtuvo una mayor incidencia de en el frente no dominado general excluyendo al algoritmo secuencial fue la experimentación realizada con 2-N, debido a que supero en el 65% de las instancias resueltas a el resto de las experimentaciones.

También se puede observar que en la experimentación realizada con 32-N obtuvo que en el 32% de las instancias resueltas supero a las experimentaciones realizadas con 4-N, 8-N y 16-N. continuando con este análisis también se observa que en la experimentación realizada con 4-N se obtuvo que en el 13% de las instancias resueltas obtuvo una mayor incidencia de soluciones superando a las experimentaciones realizadas con 8-N y 16-N.

Y finalmente tenemos que en las experimentaciones con 8-N y 16-N no superaron a ninguna de las experimentaciones realizadas en términos de la incidencia de soluciones. Por lo que el ranking del de las experimentaciones realizadas en términos de la incidencia de soluciones queda de la siguiente manera: I) Secuencial; II) 2-N; III) 32-N; III) 4-N; y finalmente se obtuvo un empate entre las experimentación IV) 8-N y 16-N.

Continuando con el análisis del algoritmo del algoritmo MPI-MEFD se evaluará el indicador de la aceleración mínima obtenida durante las experimentaciones realizadas, los resultados de este análisis se muestran en la Tabla 35.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.95833333	0.88461538	0.92	0.88461538	0.92
o4p25_1	0.92307692	0.92307692	0.92	0.92	0.92
o4p25_2	0.96	0.92307692	0.92307692	0.92307692	0.92307692
o4p25_3	0.92307692	0.96	0.92307692	0.92307692	0.92307692
o4p25_4	0.96	0.88888889	0.92307692	0.85714286	0.92307692
o4p25_5	0.92	0.92	0.92	0.92	0.95833333
o4p25_7	1	0.96	0.96	0.96	0.96
o4p25_8	0.92	0.92	0.92	0.92	0.92
o4p25_9	0.96	0.96	0.96	0.96	0.96
o4p25_10	0.96	0.92307692	0.96	0.92307692	0.96
o4p25_11	1	1	1	0.96296296	1
o4p25_12	0.96	1	0.92307692	0.96	0.96
o4p25_13	1	1	1	0.96153846	0.96153846
o4p25_14	0.96153846	0.96153846	0.96153846	0.96153846	0.96153846
o4p25_15	1	1	0.96	0.92307692	0.96
o4p25_16	0.92	0.92	0.92	0.95833333	0.95833333
o4p25_17	0.96	1	0.96	0.96	0.96
o4p25_18	1	0.96	0.96	0.92307692	0.96
o4p25_19	0.96	0.96153846	0.92307692	0.92307692	0.92307692
o4p25_20	0.96	0.96	0.96	0.96	0.96153846
o9p100_0	1.02912621	0.79699248	1.01941748	1.02941176	1.05
o9p100_1	1.03846154	1.02857143	1.01904762	1.02857143	1.05882353
o9p100_2	1.05309735	1.05309735	1.03539823	1.03539823	1.07272727

Tabla 34.- Indicador Aceleración Mínima del algoritmo MPI-MEFD.

Como se puede observar en la Tabla 35 la experimentación realiza que tuvo un mejor comportamiento en este indicador fue la realizada con 2-N, ya que esta experimentación se obtuvo que en el 33% de las instancias resueltas en algunas de estas experimentaciones al menos obtuvo el mismo tiempo de ejecución que el algoritmo secuencial.

Seguido de las experimentaciones realizadas con 4-N ya que en estas experimentaciones obtuvo que en el 29% de las instancias resueltas al menos alcanzo se obtuvo el tiempo de ejecución comparado con el algoritmo secuencial.

En tercer lugar quedan las experimentaciones realizadas con 8-N ya que obtuvo que en el 21% de las instancias resueltas al menos igualo el tiempo de ejecución comparada con el algoritmo secuencial.

Continuando con este análisis se procede a evaluar el indicador de aceleración máxima obtenida a lo largo de las experimentaciones realizadas al algoritmo MPI-MEFD, los resultados derivados de este análisis se muestran en la Tabla 36.

Instancia	2	4	8	16	32
o4p25_0	1.04166667	1.04347826	1.04166667	1.04166667	1
o4p25_1	1.04166667	1.04166667	1.08333333	1.04166667	1.08333333
o4p25_2	1.04166667	1.08333333	1.04	1.04166667	1.04166667
o4p25_3	1.04166667	1.04166667	1.04166667	1	1.04166667
o4p25_4	1.04166667	1.08333333	1.04166667	1.04166667	1.04166667
o4p25_5	1	1.04347826	1.04166667	1.04347826	1.04347826
o4p25_7	1.25	1.2	1.2	1.25	1.25
o4p25_8	1.04347826	1.04347826	1.04347826	1.04166667	1.04347826
o4p25_9	1.24	1.29166667	1.24	1.24	1.29166667
o4p25_10	1.125	1.08333333	1.125	1.08333333	1.08333333
o4p25_11	1.16	1.16	1.16666667	1.12	1.16
o4p25_12	1.29166667	1.25	1.25	1.29166667	1.25
o4p25_13	1.28	1.28	1.28	1.23076923	1.23076923
o4p25_14	1.2	1.25	1.2	1.2	1.2
o4p25_15	1.16666667	1.2173913	1.16666667	1.16666667	1.16666667
o4p25_16	1.30434783	1.25	1.25	1.30434783	1.30434783
o4p25_17	1.26086957	1.2173913	1.26086957	1.26086957	1.2173913
o4p25_18	1.26086957	1.20833333	1.20833333	1.16	1.20833333
o4p25_19	1.33333333	1.33333333	1.28	1.33333333	1.29166667
o4p25_20	1.375	1.375	1.375	1.375	1.33333333
o9p100_0	1.16666667	1.05309735	1.15533981	1.17821782	1.2020202
o9p100_1	1.14423077	1.14423077	1.14423077	1.14423077	1.17821782
o9p100_2	1.17272727	1.16216216	1.16216216	1.17272727	1.20560748

Tabla 35.- Indicador Aceleración Máxima del algoritmo MPI-MEFD.

Como se puede observar en la Tabla 36 en al menos en una de las experimentaciones realizadas con 2-N, 4-N, 8-N, 16-N y 32-N, se disminuyó el tiempo de ejecución comparado con las experimentaciones realizadas del algoritmo secuencial.

Continuando con este análisis se procede a evaluar el indicador de la aceleración promedio obtenida durante las experimentaciones realizadas, este análisis se muestra en la Tabla 37.

Instancia	2	4	8	16	32
o4p25_0	0.99086022	0.98343313	0.97944582	0.96684864	0.97149711
o4p25_1	0.98864351	0.98595533	0.97842432	0.96720017	0.97356079
o4p25_2	0.98998346	1.00222544	0.97741935	0.98651365	0.9929239
o4p25_3	0.9914847	0.99913565	0.9801498	0.97980149	0.98791563
o4p25_4	0.98870141	0.98257789	0.97489247	0.96542065	0.97385856
o4p25_5	0.98182796	0.99787284	0.97671216	0.98086129	0.98312295
o4p25_7	1.06119107	1.06398784	1.05015715	1.04639371	1.05441328
o4p25_8	0.98339645	0.98839645	0.97796634	0.96880893	0.97398317
o4p25_9	1.0633871	1.07619682	1.05370968	1.05255859	1.06388499
o4p25_10	1.03096361	1.02702647	1.02047146	1.00738213	1.01596361
o4p25_11	1.07062862	1.06804797	1.0640488	1.04537451	1.05230769
o4p25_12	1.06677419	1.07333333	1.05435484	1.05446237	1.0594086
o4p25_13	1.09349462	1.08659222	1.08257191	1.06372484	1.07196765
o4p25_14	1.02934657	1.02692721	1.01885443	1.00830438	1.01352771
o4p25_15	1.0618864	1.055041	1.04935484	1.03742457	1.04282375
o4p25_16	1.02149135	1.03377407	1.00991585	1.01295933	1.01968911
o4p25_17	1.07789855	1.0787798	1.05895278	1.06063815	1.06606358
o4p25_18	1.05698223	1.04986209	1.04320009	1.03077085	1.03810659
o4p25_19	1.07978495	1.07429694	1.06858974	1.05641026	1.05794458
o4p25_20	1.13032258	1.12678246	1.11624069	1.10313069	1.11259053
o9p100_0	1.07083704	0.88931948	1.06543551	1.07730469	1.10617227
o9p100_1	1.07355727	1.07129339	1.07125729	1.07091121	1.0999633
o9p100_2	1.09190338	1.09602956	1.09129467	1.0923639	1.12140212

Tabla 36.- Indicador Aceleración Promedio del algoritmo MPI-MEFD.

Como se puede observar en la Tabla 37 la experimentación que tuvo un mejor comportamiento en este indicador fue la experimentación realizada con 2-N ya que esta supero al resto de las experimentaciones en el 48% de las instancias resueltas.

Por otro lado tenemos que las experimentaciones realizadas con 4-N se obtuvo que en el 39% de las instancias resueltas estas experimentaciones obtuvieron las mejores aceleraciones. Además se puede observar que en las experimentaciones realizadas con 32-P obtuvieron una mayor aceleración en el 13% de las instancias.

Continuando con este análisis se procede a evaluar el indicador de la varianza obtenida por la aceleración, los resultados obtenidos de este análisis se muestran en la Tabla 38.

Instancia	2-P	4-P	8-P	16-P	32-P
o4p25_0	0.00062256	0.00111536	0.00091465	0.00120708	0.00053151
o4p25_1	0.00084977	0.00066948	0.00102985	0.00109829	0.00136117
o4p25_2	0.00071868	0.00140327	0.00075807	0.00125408	0.0009652
o4p25_3	0.00090283	0.00080029	0.00087272	0.00057366	0.00116773
o4p25_4	0.00086276	0.00116111	0.00094384	0.00140301	0.00113823
o4p25_5	0.00071082	0.00088568	0.00070276	0.00104798	0.00051629

o4p25_7	0.00299348	0.00280511	0.00307642	0.00380323	0.00332652
o4p25_8	0.00146823	0.00100309	0.00095314	0.00069828	0.00081278
o4p25_9	0.00390618	0.00421302	0.0040478	0.00390776	0.00500318
o4p25_10	0.0015224	0.00162794	0.00194666	0.00163862	0.00113589
o4p25_11	0.00152559	0.00178024	0.00176289	0.00138697	0.00239893
o4p25_12	0.00495644	0.00389803	0.00476726	0.00525545	0.00408612
o4p25_13	0.00385221	0.00387741	0.00400137	0.00394755	0.00422807
o4p25_14	0.00191146	0.0026054	0.00206899	0.00198586	0.0021168
o4p25_15	0.00244745	0.00248326	0.00244743	0.00297914	0.00282593
o4p25_16	0.00393283	0.00291463	0.0028684	0.0036158	0.00335881
o4p25_17	0.00559011	0.00441998	0.00479904	0.00522035	0.00469227
o4p25_18	0.00311801	0.00269225	0.00315153	0.00262598	0.00380695
o4p25_19	0.00616823	0.00578773	0.00556649	0.00616071	0.00648515
o4p25_20	0.00962122	0.01159074	0.01026364	0.00975435	0.00894042
o9p100_0	0.00117796	0.00339978	0.00111804	0.00119196	0.00129721
o9p100_1	0.00073893	0.00086893	0.00080086	0.0007717	0.00082876
o9p100_2	0.00101761	0.00106772	0.00103722	0.00131475	0.00111535

Tabla 37.- Indicador de la varianza de la Aceleración del algoritmo MPI-MEFD.

Como se puede observar en la Tabla 38 la varianza obtenida por la aceleración no representa un impacto significativo a la aceleración promedio obtenida a lo largo de las experimentaciones realizadas.

En términos generales se puede decir que durante las experimentaciones realizadas no en todas las instancias de 4 objetivos y 25 proyectos se pudo disminuir el tiempo de ejecución comparado con el algoritmo secuencial.

7.4.5. Comentarios derivados de la experimentación de los Algoritmos implementados.

En términos generales se observó que en las experimentaciones realizadas a los algoritmos implementados que hacen uso de la librería OMP concluyo que existió una mayor incidencia de soluciones en el frente no dominado general en algunas de las instancias resueltas.

En la Tabla 39 se muestran las experimentaciones realizadas a los algoritmos OMP-GG y OMP-FD que superaron la incidencia de soluciones en el frente cero general.

OMP-FD	OMP-GG
2-P	2-P
32-P	4-P
	32-P

Tabla 38.- Experimentaciones que superaron en calidad de solución a la versión secuencial.

Como se puede observar en la Tabla 39 las en las dos experimentaciones realizadas con el algoritmo OMP-FD que tuvieron un mejor comportamiento en término de la incidencia de soluciones por lo que se procedió a evaluar en términos de la aceleración y se determinó que el algoritmo que tuvo una mayor aceleración fue la experimentación realizada con 2-P.

Debido a que la experimentación realizada con 2-P obtuvo una aceleración promedio mayor en todas las instancias

Por otro lado tenemos que en las experimentaciones realizadas con 2-P, 4-P, y 32-P del algoritmo OMP-GG superaron en términos de la incidencia de soluciones al algoritmo secuencial, al contrastar estas experimentaciones con el ranking obtenido en el análisis de la aceleración se obtuvo que la experimentación que tuvo el mejor comportamiento en términos de incidencia de soluciones obtuvo el segundo lugar en términos de la aceleración.

También se observó que la experimentación realizada con 4-N fue la experimentación que presento una mayor aceleración el que obtuvo una mejor aceleración. Y finalmente la experimentación realizada con 32-P obtuvo el penúltimo lugar en términos de la aceleración.

Por lo que le da soporte a que los las experimentaciones realizadas con 2-P y 4-P del algoritmo OMP-GG junto la experimentación realizada con 2-P del algoritmo OMP-FD son las experimentaciones que tuvieron los mejores comportamientos.

Por otro lado tenemos que los algoritmos que hacen uso de la librería de MPI no pudieron mejorar la calidad de las soluciones generadas e inclusive las soluciones generadas perdieron calidad de soluciones.

Ya que únicamente permanecieron solo las soluciones resultantes del algoritmo secuencial, pero la calidad de las soluciones obtenidas por las experimentaciones de los algoritmos paralelos son buenas ya que el frente cero no dominado general es pequeño ya que el tamaño de la población oscila entre 1 y 9 individuos por lo que se puede decir que los algoritmo paralelos generaron buenas soluciones.

Por lo que tenemos que en las experimentaciones realizadas algoritmo MPI-GG quedo de la siguiente manera: I) 2-N II) 8-N y finalmente IV) 16-N; debido a que solamente se obtuvieron los tres mejores por que las experimentaciones realizadas con 4-N y 32-N solamente obtuvieron una incidencia mayor en el 4% de las instancias resueltas.

Comparando este ranking con el obtenido en términos de la aceleración se tenemos que el algoritmo que tuvo una mejor aceleración fue uno de los peores algoritmos en términos de la incidencia de soluciones en el frente no dominado general.

Pero por otro lado tenemos que las experimentaciones realizadas con 8-N fue el que obtuvo el segundo lugar en términos de la incidencia de soluciones en el frente no dominado general, así como también en términos de la aceleración.

Y finalmente tenemos que en las experimentaciones realizadas con 16-N fue el algoritmo con uno de los peores comportamientos en términos de la aceleración, por lo que con las experimentaciones realizadas al algoritmo MPI-MEGG le da soporte a que el algoritmo que tuvo un mejor comportamiento fue a las experimentaciones realizadas con 8-N.

Analizando el algoritmo MPI-MEFD tenemos que el ranking obtenido en calidad de solución fue el siguiente: 2-N; III) 8-N; IV) 16-N; ya que fueron las experimentación que se obtuvo a lo menos que en el 10% de instancias resueltas obtuvieron una mayor incidencia de soluciones en el frente no dominado.

Además tenemos que durante las experimentaciones realizadas al algoritmo MPI- MEFD no se pudo alcanzar una aceleración en todas las instancias resueltas, por lo que en términos generales podemos decir que el algoritmo que tuvo un mejor comportamiento a lo largo de las experimentaciones fue el Algoritmo MPI-MEGG en la experimentación realizada con 8-N.

Basado en las experimentaciones que se realizaron se procedió a realizar la prueba estadística de Friedman y la prueba post-hoc de Bonferroni-Dunn las cuales tiene como finalidad demostrar que existe una diferencia significativa entre las experimentaciones que tienen los mejores comportamientos observados en el análisis anterior. Las experimentaciones a analizar son las siguientes:

- Secuencial
- OMP-GG-2P
- OMP-GG-4P
- OMP-FD-2P
- MPI-GG-2N
- MPI-GG-8N.

Por lo que se formaron 6 grupos donde cada uno de los grupos representan cada una de las experimentaciones a realizar, cada uno de estos grupos contiene 24 elementos lo cuales representan cada una de las instancias durante las ejecuciones.

La hipótesis nula que se pretendo rechazar es que la media de las aceleraciones alcanzadas durante las experimentaciones realizadas son iguales. Ya que lo que se busca es que exista una diferencia significativa entre las aceleraciones obtenidas.

Al realizar la prueba e Friedman se obtuvo el que el rango promedio fue de 344.30396 también se obtuvo que la hipótesis es se rechaza por lo que concluye que existe diferencia significativa en la aceleración obtenida en las experimentaciones realizadas.

Continuando con este análisis de la validación estadística se muestra el ranking obtenido por la prueba de Friedman, el ranking se muestra en la Tabla 40.

Rango Promedio	Experimentación
1.30435	Secuencial
1.69565	OMP-FD-2P
3.52174	MPI-GG-2N
3.52174	OMP-GG-2P
4.95652	MPI-GG-8N
6.00000	OMP-GG-4P

Tabla 39.- Ranking obtenido por la prueba de Friedman.

Como se puede mostrar en la Tabla 40 tenemos que en las experimentaciones realizadas con el algoritmo secuencial fue la que obtuvo un rango promedio más bajo pero en esta experimentación sirve como punto de comparación para calcular la aceleración obtenida de las demás experimentaciones. Por lo que en esta experimentación no existe aceleración.

Continuando con el análisis de la validación estadística se procede a evaluar los resultados obtenidos de la prueba post-hoc de Bonferroni-Dunn, en las cuales se utilizaron las experimentaciones realizadas al algoritmo secuencial como método de comparación. Obteniendo los resultados que se muestran en la Tabla 40.

Comparación	P-Valor Ajustado	Resultado
Secuencial vs OMP-GG-4P	0.00000	H0 es rechazada

Comparación	P-Valor Ajustado	Resultado
Secuencial vs MPI-MEGG-8N	0.00000	H0 es rechazada
Secuencial vs OMP-GG-2P	0.00029	H0 es rechazada
Secuencial vs MPI-MEGG-2N	0.00029	H0 es rechazada
Secuencial vs OMP-FD-2P	1.00000	H0 es aceptada.

Tabla 40.- prueba post-hoc de Bonferroni-Dunn.

Como se puede observar en la Tabla 40 no indica bajo esta prueba la única comparación que acepta la hipótesis nula es la realizada entre las experimentaciones secuencial vs OMP-FD-2P.

Con relación a los resultados obtenidos se procede a implementar un algoritmo paralelo híbrido que haga uso de la combinación de los patrones de diseño que obtuvieron una mejor varianza significativa.

En este caso tenemos que los algoritmos que previeron una mayor varianza significativa en términos de la aceleración fueron MPI-MEGG y OMP-GG en las experimentaciones realizadas con 2-N y 2-P respectivamente.

La implementación de este algoritmo lo que busca es disminuir aún más el tiempo de ejecución y además intentar mejorar la calidad de las soluciones generadas, en virtud de que el algoritmo MPI-MEGG con las experimentaciones realizadas con 8N obtienen los mejores tiempos de ejecución pero la calidad de soluciones disminuyo comparado con la versión secuencial.

Por otro lado tenemos que en términos de la incidencia de soluciones en el frente no dominado el algoritmo OMP-GG y OMP-FD obtuvo una mayor incidencia de soluciones en el frente no dominado.

7.4.6. Análisis del algoritmo MEGGFD.

Para realizar el análisis de este patrón de diseño realizara se realizarán las experimentaciones con 2-N y 2-P con lo cual se analizara la incidencia de soluciones en el frente no dominado general, los resultados de este análisis se muestra en la Tabla 42.

Instancia	S	MPI-MEGG-2N	OMP-FD-2P	Hibrido
o4p25_0	100%	0%	0%	0%
o4p25_1	0%	0%	0%	100%
o4p25_2	50%	0%	0%	50%
o4p25_3	100%	0%	0%	0%

o4p25_4	0%	0%	0%	100%
o4p25_5	100%	0%	0%	0%
o4p25_7	100%	0%	0%	0%
o4p25_8	75%	0%	0%	25%
o4p25_9	100%	0%	0%	0%
o4p25_10	0%	0%	0%	100%
o4p25_11	67%	0%	0%	33%
o4p25_12	100%	0%	0%	0%
o4p25_13	0%	0%	0%	100%
o4p25_14	100%	0%	0%	0%
o4p25_15	33%	0%	0%	67%
o4p25_16	75%	0%	0%	25%
o4p25_17	63%	0%	0%	38%
o4p25_18	80%	0%	0%	20%
o4p25_19	75%	0%	0%	25%
o4p25_20	67%	0%	0%	33%
o9p100_0	100%	0%	0%	0%
o9p100_1	0%	0%	0%	100%
o9p100_2	0%	0%	0%	100%

Tabla 41.- Frente no dominado del algoritmo híbrido.

Como se puede observar en la Tabla 42 se observó que existió una mejora en la incidencia de soluciones en el frente no dominado en las experimentaciones realizadas al algoritmo Híbrido, mejorando las incidencias obtenidas por las experimentaciones realizadas al algoritmo OMP-FD con 2-P y MPI-MEEGG con 2-N.

Por otro lado tenemos que en el 15% de las instancias resueltas las experimentaciones realizadas con 2-P y 2-N que en el 62% de las soluciones no obtuvo ninguna solución en el frente no dominado, pero tenemos que en el 37.5% de las instancias resueltas que mantuvo una incidencia de soluciones no dominado.

Además también se observó que en el 33% de las instancias resueltas este algoritmo obtuvo una mayor incidencia de soluciones en el frente no dominado supero al algoritmo secuencial este algoritmo aumento la incidencia de soluciones en el frente no dominado, también se observó que en 25% de las instancias resueltas el algoritmo secuencias no obtuvo ninguna incidencia de soluciones en el frente no dominado.

Continuando con este análisis se procede a evaluar la aceleración obtenida con este algoritmo haciendo uso del indicador aceleración mínima, los resultados obtenidos se muestran en la Tabla 43.

Instancia	OMP-FD-2-P	MPI-MEGG-2-N	H
o4p25_0	0.958333333	1.49	3.125
o4p25_1	0.92307692	1.5	2.27272727
o4p25_2	0.96	1.5	1.71428571
o4p25_3	0.92307692	1.5	1.33333333
o4p25_4	0.96	1.51	1.18181818
o4p25_5	0.92	1.58	0.86666667
o4p25_7	1	1.78	0.70588235
o4p25_8	0.92	1.43	0.64864865
o4p25_9	0.96	1.52	0.58536585
o4p25_10	0.96	1.49	0.53333333
o4p25_11	1	1.61	0.52
o4p25_12	0.96	1.48	0.44444444
o4p25_13	1	1.57	0.44827586
o4p25_14	0.96153846	1.51	0.3968254
o4p25_15	1	1.57	0.35820896
o4p25_16	0.92	1.59	0.32857143
o4p25_17	0.96	1.61	0.32
o4p25_18	1	1.63	0.30379747
o4p25_19	0.96	1.51	0.29069767
o4p25_20	0.96	1.77	0.27906977
o9p100_0	1.02912621	1.8	0.77372263
o9p100_1	1.03846154	1.97	0.66257669
o9p100_2	1.05309735	1.91	0.7375

Tabla 42.- Indicador de Aceleración Mínima del algoritmo Híbrido.

Como se puede observar en la Tabla 23 las experimentaciones realizadas al algoritmo MPI-MEGG con 2-N ya que en el 96% de las instancias al menos en alguna de sus experimentaciones se alcanzó una aceleración.

Además se observa las experimentaciones realizadas a las experimentaciones realizadas al algoritmo OMP-FD ya que en estas experimentaciones se obtuvo que al menos en algunas de estas se alcanzó una aceleración.

Por otro lado tenemos que las experimentaciones realizadas al algoritmo Híbrido implementado se obtuvo que el 21% de las instancias resueltas existió al menos una aceleración en alguna de estas experimentaciones.

Continuando con este análisis se procede a evaluar el indicador de la aceleración maximiza obtenida a lo largo de las experimentación, estos resultados se muestran en la Tabla 44.

Instancia	OMP-FD-2-P	MPI-MEGG-2-N	H
o4p25_0	1.041666667	3.37	7.25
o4p25_1	1.041666667	3.47	4.28571429
o4p25_2	1.041666667	3.32	2.58333333
o4p25_3	1.041666667	3.35	2

o4p25_4	1.041666667	4.47	1.5
o4p25_5	1	2.46	1.38095238
o4p25_7	1.25	3.95	1
o4p25_8	1.043478261	3.15	0.87878788
o4p25_9	1.24	3.44	0.78947368
o4p25_10	1.125	3.55	0.64285714
o4p25_11	1.16	5.66	0.61702128
o4p25_12	1.291666667	3.79	0.60784314
o4p25_13	1.28	7.33	0.58181818
o4p25_14	1.2	3.43	0.50847458
o4p25_15	1.166666667	3.65	0.44444444
o4p25_16	1.304347826	3.29	0.44776119
o4p25_17	1.260869565	3.66	0.4084507
o4p25_18	1.260869565	3.17	0.38157895
o4p25_19	1.333333333	4.19	0.39506173
o4p25_20	1.375	3.85	0.37647059
o9p100_0	1.166666667	8.37	1.00847458
o9p100_1	1.144230769	8.15	0.84057971
o9p100_2	1.172727273	9.23	0.92805755

Tabla 43.-Indicador de Aceleración Máxima del algoritmo Híbrido.

Como se puede observar en la Tabla 44 las experimentaciones realizadas al algoritmo MPI-MEEGG son las que obtuvieron un mejor comportamiento en este indicador ya que en el 91.6% de las instancias supero al resto de las experimentaciones.

Además se puede observar que en el .83% alcanzo de las instancias alcanzo una mayor aceleración, superando solo a las experimentaciones realizadas por el algoritmo OMP-FD. Se procederá a evaluar el indicador de aceleración promedio, los resultados obtenidos de este indicador se muestran en la Tabla 45.

Instancia	OMP-FD-2-P	MPI-MEGG-2-N	H
o4p25_0	0.990860215	1.34	4.98798003
o4p25_1	0.988643507	1.36	2.92890541
o4p25_2	0.989983457	1.36	1.99241243
o4p25_3	0.991484698	1.37	1.58729293
o4p25_4	0.988701406	1.36	1.3449637
o4p25_5	0.981827957	1.39	0.9527809
o4p25_7	1.061191067	1.36	0.81436395
o4p25_8	0.983396447	1.31	0.71760666
o4p25_9	1.063387097	1.37	0.65375312
o4p25_10	1.030963606	1.36	0.57319949
o4p25_11	1.070628619	1.41	0.56062402
o4p25_12	1.066774194	1.33	0.49937398
o4p25_13	1.093494624	1.35	0.48444221
o4p25_14	1.029346567	1.38	0.42012401

o4p25_15	1.061886396	1.41	0.39610888
o4p25_16	1.021491351	1.33	0.35283598
o4p25_17	1.077898551	1.35	0.35375774
o4p25_18	1.056982235	1.38	0.32658769
o4p25_19	1.079784946	1.33	0.32123436
o4p25_20	1.130322581	1.4	0.31469203
o9p100_0	1.070837044	1.74	0.86883504
o9p100_1	1.073557268	1.81	0.73559548
o9p100_2	1.091903375	1.84	0.82791605

Tabla 44.- Indicador de Aceleración Promedio del algoritmo Híbrido.

Como se puede observar en la Tabla 44 las experimentaciones realizadas al algoritmo MPI-MEGG fue el que obtuvo una mejor aceleración promedio ya que en el 83% de las instancias supero al resto de las experimentaciones realizadas con los algoritmos OMP-FD y el Híbrido.

Además tenemos que las experimentaciones realizadas al algoritmo híbrido alcanzo a obtener la mejor aceleración promedio en el 13% de las instancias resueltas, finalmente tenemos que en este indicador las experimentaciones realizadas al algoritmo secuencial fue el que obtuvo las peores aceleraciones promedio.

Continuando con el análisis de las experimentaciones realizadas se procede a evaluar el indicador de la varianza, los resultados obtenidos en este indicador se muestran en la Tabla 46.

Instancia	OMP-FD-2-P	MPI-MEGG-2-N	H
o4p25_0	0.000622558	0.008	0.94135491
o4p25_1	0.000849769	0.006	0.20735664
o4p25_2	0.000718681	0.004	0.04181831
o4p25_3	0.000902831	0.002	0.02163713
o4p25_4	0.000862762	0.004	0.00744698
o4p25_5	0.000710816	0.006	0.00335646
o4p25_7	0.002993483	0.01	0.00391447
o4p25_8	0.001468234	0.008	0.00279698
o4p25_9	0.00390618	0.005	0.00180548
o4p25_10	0.001522397	0.004	0.00066924
o4p25_11	0.001525592	0.006	0.00050627
o4p25_12	0.00495644	0.003	0.00103712
o4p25_13	0.003852214	0.006	0.00082459
o4p25_14	0.001911457	0.003	0.00037437
o4p25_15	0.002447451	0.007	0.00041838
o4p25_16	0.003932829	0.008	0.00038686
o4p25_17	0.005590106	0.005	0.00066915
o4p25_18	0.00311801	0.008	0.00035034
o4p25_19	0.006168233	0.005	0.00048663
o4p25_20	0.009621222	0.009	0.00065524
o9p100_0	0.001177956	0.001	0.00338503

o9p100_1	0.000738926	0.002	0.00235776
o9p100_2	0.001017605	0.001	0.00232417

Tabla 45.- Indicador de la varianza obtenida de la Aceleración del algoritmo Híbrido.

Como se puede observar en la Tabla 46 se observó que la varianza obtenida es muy pequeña a excepción de las instancias o4p25_0, o4p25_1, o4p25_2, o4p25_3, o4p25_4 y o4p25_5 que fueron obtenidas por el algoritmo híbrido. Estas instancias coinciden con las mayores aceleraciones alcanzadas por estas experimentaciones.

El término general tenemos que el algoritmo híbrido implementado se alcanzó a mejorar la calidad de las soluciones ya que comparado con los algoritmos paralelos que tuvieron una mayor incidencia de soluciones observadas en las experimentaciones anteriores el algoritmo híbrido generó mejores soluciones.

Por lo que al generar el nuevo frente general de las soluciones con los algoritmos secuencial, OMP-FD, MPI-GG y el híbrido solamente permanecieron las soluciones generadas durante las experimentaciones de los algoritmos Secuencial y el híbrido.

Pero al evaluar en términos de la aceleración el algoritmo que tuvo una mayor aceleración y obteniendo buenas soluciones fue el algoritmo MPI-MEGG en las experimentaciones realizadas con 2-N.

8. Conclusiones.

Basado en las experimentaciones a los algoritmos secuenciales se observó que el perfil preferencial utilizado para resolver las instancias no solo obtuvo las mejores soluciones con mejor compromiso sí que también ayudaron a aproximar la ROI del DM.

Debido a que basada en la comparación de soluciones en la incidencia de soluciones en el frente no dominado, los algoritmos con incorporación de preferencia NO-ACO y NSGA obtuvieron una mayor incidencia en comparación con los algoritmos sin incorporación de preferencias NS-ACO y NSGA-II. Donde se observó que el algoritmo que se vio más beneficiado por el perfil preferencial fue el algoritmo NO-ACO ya que la incidencia de soluciones fue mucho mayor que las obtenidas por su versión secuencial.

Después de analizar la incidencia de soluciones en el frente no dominado, se procedió a hacer el análisis de la frontera No Superada Estrictamente lo cual como era de esperarse los algoritmos que incorporan preferencia fueron los que alcanzaron una mayor incidencia de soluciones en esta frontera.

El algoritmo que obtuvo una mayor incidencia de soluciones en el frente no dominado fue el algoritmo NOACO, por lo que fue el algoritmo que se paralelizó con la finalidad de mejorar los tiempos de ejecución y en calidad de solución.

Para determinar cuál de los algoritmos obtuvo una mejoría en tiempo de ejecución se hizo uso de la métrica de la aceleración, por el lado del análisis de la calidad de soluciones se hizo bajo el criterio de las soluciones en la frontera no dominada estrictamente y el frente no dominado.

Por lo que el objetivo de implementar los algoritmos paralelos es disminuir los tiempos de ejecución e intentar mejorar la calidad de las soluciones obtenidas por la versión secuencial del algoritmo NO-ACO.

La aplicabilidad de las técnicas de diseño de software (Patrones de Diseño) resulto muy intuitiva ya que existen muchas pautas para la implementación de meta heurísticas paralelas, por lo que la implementación resulto sencilla e intuitiva.

Por lo que se procedió a paralelizar la Meta-heurística NO-ACO implementado dos algoritmo que hacen uso de memoria compartida y dos que hacen uso de memoria distribuida, los cuales fueron implementados utilizando diferentes Patrones de Diseño.

Los patrones utilizados para las implementaciones que hacen uso de la librería OMP (memoria compartida) fueron el Flujo de Datos (algoritmo denominado OMP-FD) y el de Grano Grueso (algoritmo denomino OMP-GG).

Por otro lado tenemos que para implementar los algoritmos que hacen uso de la librería MPI (memoria distribuida) se realizó una hibridación de patrones de diseño en los cuales se combinaron el patrón de Maestro Esclavo y Grano Grueso (Algoritmo denominado MPI-MEGG) para la implementación de un algoritmo paralelo.

Para la implementación del segundo algoritmo paralelo se utilizaron los patrones de diseño Maestro Esclavo y flujo de datos (Algoritmo denominado MPI-MEFD). Para el análisis de estos algoritmos se realizaron experimentaciones con 2-P, 4-P, 8-P, 16-P y 32-P (donde P representa el número de hilos de procesamiento) para las implementaciones que hacen uso de la librería de OMP.

Por otro lado tenemos para el análisis de los algoritmos implementados con la librería de MPI se hicieron experimentaciones con 2-N, 4-N, 8-N, 16-N y 32-N (donde N representa el número de procesadores utilizados).

En términos generales se observó que los algoritmos implementados con la librería OMP alcanzaron a tener una aceleración e inclusive alcanzaron a superar la calidad de soluciones a las obtenidas por la versión secuencial del algoritmo en al menos el 30% de las instancias resueltas.

También se observó que las experimentaciones realizadas con 8-P, 16-P y 32-P no alcanzaron una aceleración promedio en todas las instancias, por otro lado se observó que en las experimentaciones realizadas con 8-P en algunas instancias se alcanzó una aceleración.

Finalmente se puede concluir los algoritmos que tienen un mejor comportamiento en términos de calidad de solución y de la aceleración fueron las experimentaciones realizadas con 2-P en los dos algoritmos que hacen uso de la librería OMP.

Durante el análisis de los algoritmos que hacen uso de la librería MPI se observó que en términos de la calidad de solución el algoritmo secuencial obtuvo mejores resultados que las experimentaciones realizadas con los algoritmos paralelos.

Ya que integrar generar el frente no dominado general solamente permanecieron soluciones del algoritmo secuencial, pero este frente es muy pequeño ya que el número de soluciones que inciden oscila entre 1 y 10 soluciones. Lo que indica que la calidad de soluciones obtenidas por los algoritmos

implementados que hacen uso de la librería MPI obtiene soluciones con una buena calidad ya que fueron dominadas por muy pocas soluciones del algoritmo secuencial.

Por lo que se realizó una nueva construcción de un frente no dominado en el cual se excluyeron las soluciones generadas por el algoritmo secuencial, en este frente se observó que las mejores dos experimentaciones fueron las realizadas con 2-N y 8-N.

Pero el por otro lado tenemos que los algoritmos implementados que hacen uso de MPI fueron los que alcanzaron mayor aceleración promedio, con el análisis del algoritmo MPI-MEGG fue la que obtuvo una mejor aceleración promedio.

Además se pudo observar en el análisis del algoritmo MPI-MEGG empezó a decaer la aceleración partir de las experimentaciones realizadas con 8-N, aunque en esta experimentación se pudo mejorar los tiempos de ejecución en todas las instancias.

La experimentación que obtuvo mejores resultados en términos de la calidad de solución y aceleración fue la experimentación realizada con 2-N ya que fue el algoritmo que obtuvo una mejor incidencia de soluciones en el frente no dominado y alcanzo una buena aceleración.

En el análisis del algoritmo MPI-MEFD se observó que en las experimentaciones realizadas con 4-N supero en el 48% de las instancias términos de la aceleración al resto de las experimentaciones realizadas, siendo esta la experimentación con una mejor aceleración.

Por lo que posteriormente se procedió realizar la validación estadística de Friedman, en las cuales se analizaron los tiempos de ejecución de las mejores experimentaciones, obteniendo 5 conjuntos de datos que representan las mejores experimentaciones realizadas.

Las cuales contienen 24 elementos donde cada uno de los elementos los tiempos de ejecución promedio obtenidas en cada una de las instancias resueltas. La hipótesis nula que se persigue es que exista una diferencia significativa entre las medias de los tiempos de ejecución (H_0).

Como resultado de la prueba de Friedman se obtuvo que si existe una diferencia significativa en los tiempos de ejecución obtenidos entre los tiempos de ejecución del algoritmo secuencial y el algoritmo paralelos implementados.

Por lo que se implementó un algoritmo paralelo el cual se implementó con los patrones de diseño que reportaron una mayor diferencia significativa en términos del tiempo de ejecución por lo que se implementó el algoritmo híbrido (H-MEGGFD).

En este algoritmo se alcanzó una mejoría en términos de la calidad de la solución ya que obtuvo una incidencia mayor de soluciones en el frente no dominado en comparación con el resto de los algoritmos comparado con las experimentaciones que obtuvieron mejores resultados.

Además se observó que el resto de los algoritmos su incidencia fue nula en el frente no dominado, finalmente para concluir con el análisis del algoritmo H-MEGGFD fue el segundo mejor en términos de la calidad de solución. Pero en términos de la aceleración fue el algoritmo MPI-GG obtenido en términos de la aceleración.

9. Referencias

- Alejandro Arito, F. L. (2010). Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados, 83. Retrieved from <http://www0.unsl.edu.ar/~dmcc/files/tesis-f.pdf>
- Coello, C. A. C. (2000). Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. *Congress on Evolutionary Computation. CEC2000.*, 30–37. <https://doi.org/10.1109/CEC.2000.870272>
- Coello, C. A. C. (2008). Optimización en Ingeniería El Método de Patrones de Búsqueda de Hooke-Jeeves, (2508).
- Cortés, A. (n.d.). Estructura híbrida MPI-OpenMP para predicción de incendios forestales.
- De la fuente, A. la fuente. (2005). Algoritmos genéticos paralelos.
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (n.d.). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II.
- Duarte Muñoz, A., & Pantrigo Fernandez, Juan Jose Gallego Carrillo, M. (2007). Metaheurísticas. *Dykinson*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Eduardo, C., & Carvalho, D. De. (2011). Aplicación del método Electre III en la clasificación de clústeres de artesanías, 7(1), 97–112.
- Fern, A., & Toro, F. De. (n.d.). Paralelización de Algoritmos Multi-objetivo: Aplicación al Transporte de Mercancías.
- Fernández, E. (n.d.). Modelo y sistema de apoyo a la decisión para problemas de cartera de proyectos con relevancia social, X.
- Fernández, E., Lopez, E., Mazcorro, G., Olmedo, R., & Coello Coello, C. A. (2013). Application of the non-outranked sorting genetic algorithm to public project portfolio selection. *Information Sciences*, 228, 131–149. <https://doi.org/10.1016/j.ins.2012.11.018>
- Figueira, J., Greco, S., & Ehrogott, M. (2005). Multiple Criteria Decision Analysis: State of the Art Surveys, 78. <https://doi.org/10.1007/b100605>
- Fülöp, J. (2001). Introduction to Decision Making Methods. *Operations Research*, 1–15. <https://doi.org/10.1.1.86.6292>
- Genis-triana, C., & Rivera-lópez, A. R. R. (2010). Algoritmos paralelos de codificación de video basado en H. 264 / AVC con balanceo predictivo de carga, 2(2).
- González, E. F., & Castillo, J. N. (2001). Model and decision support system for portfolio problems in projects of social importance. *Gestión y Política Pública*.
- González, E. F., Cervantes, E. L., Castillo, J. N., & López, I. V. (2011). Application of Multi-Objective Metaheuristics to Public Portfolio Selection Through Multidimensional Modelling of Social Return. *Gestion y Politica Publica*, 20(2), 381–432.
- Kathavate, S., & Srinath, N. K. (2014). Efficiency of Parallel Algorithms on Multi Core Systems Using

- OpenMP, 3(10), 8237–8241.
- Mousseau, V., Slowinski, R., & Zielniewicz, P. (2000). A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. *Computers and Operations Research*, 27(7–8), 757–777. [https://doi.org/10.1016/S0305-0548\(99\)00117-3](https://doi.org/10.1016/S0305-0548(99)00117-3)
- Nesmachnow, S. (2004). Algoritmos Genéticos y Paralelos y su Aplicación al Diseño de Redes de Comunicaciones Confiables, 174. Retrieved from <http://www.fing.edu.uy/~sergion/Tesis.pdf>
- Ni, D., Carre, J., Trigos, W., Gelvez, S., Gonz, H., & Barrios, C. (2015). 3D Seismic modeling using hybrid architectures . Modelado sísmico en 3D haciendo uso de arquitecturas híbridas, (September).
- Osman, I. H., & Kelly, J. P. (1997). Meta-Heuristics Theory and Applications. *Journal of the Operational Research Society*, 48(6), 657–657. <https://doi.org/10.1057/palgrave.jors.2600781>
- Pena, R. R., Rebollo, L. P., Oliveras, K. G., & Mateu, A. V. (2007). Use and Evaluation of Electre Iii / Iv, 19. Retrieved from <http://deim.urv.cat/recerca/reports/DEIM-RT-07-003.html>
- Place, P., Lattre, D., & Cedex, P. (2005). Chapter 4 Electre Methods Manual Do Lansade. *Multiple Criteria Decision Analysis: State of the Art Surveys*.
- Rafael Arturo Trujillo Rasúa, Antonio M. Vidal Maciá, V. M. G. M. (2009). *algoritmos paralelos para la solucione de problemas de optimizacion.pdf*. Universidad Politecnica de Valencia.
- Rivera Zárate, G., Gómez Santillan, C. G., & René, F. G. E. (2015). Enfoque Meta heurístico Híbrido para el Manejo de Muchos Objetivos en Optimización de Cartera de Proyectos Interdependientes con Decisiones de Apoyo Parcial.
- Robles Algarín, C. A. (2010). Optimización por colonia de hormigas : aplicaciones y tendencias. *Revista Ingeniería Solidaria*, 6, 83–89.
- Rodríguez, R. G., Reyes, L. C., & López, F. (2010). Hiper-heurístico Aplicado al Problema de Cartera de Proyectos Sociales, 25–36.
- Sanjay Kumar Sharma, K. G. (2012). Performance analysis of parallel search algorithms on multi-core Systems using OpenMp. *The Proceedings of International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, (0).
- Santander-jim, S., Vega-rodr, M. A., Juan, A. G., & Juan, M. S. (n.d.). Un Sistema Híbrido MPI / OpenMP Basado en el Algoritmo NSGA-II para la Inferencia de Arboles Filogenéticos.
- Sapag, R. (2000). Preparación y evaluación de proyectos. 4 Edición. Retrieved from <http://www.utntyh.com/alumnos/wp-content/uploads/2013/03/Preparacion-Y-Evaluacion-De-Proyectos-Sapag-Sapag.pdf>
- Tam, C. M., Tong, T. K. L., & Lau, C. T. (2003). ELECTRE III in evaluating performance of construction plants: case study on concrete vibrators. *Construction Innovation*, 3(1), 45–61. <https://doi.org/10.1108/14714170310814846>
- Tim B. Cooper and Jeffrey H. Kingston. (1996). Complexity Issues.
- URBINA, G. B., & CASTELLANOS, M. Á. T. (2006). *Evaluación de proyectos*. McGraw-Hill (Vol. 3).

- Victoria Ruiz Martínez, C. G. G. S. (2016). *Modelado de Efectos de Primer Orden en el Problema de Cartera de Proyectos Públicos*.
- Zak, J., & Kruszyński, M. (2015). Application of AHP and ELECTRE III/IV methods to multiple level, multiple criteria evaluation of urban transportation projects. *Transportation Research Procedia*, 10(July), 820–830. <https://doi.org/10.1016/j.trpro.2015.09.035>
- Zanakis, S. H., & Evans, J. R. (1981). Heuristic “ Optimization ”: Why , When , and How to Use It WHY , WHEN , AND HOW TO USE IT. *Interfaces*, 11(5)(August 2015), 84–91.