

AUTORIZACIÓN DE USO DE DERECHOS DE AUTOR OTORGADO POR

Luis Rodolfo García Nieto, mayor de edad, con domicilio ubicado en Calle Canada #409 col. Magdalena Aguilar, en mi calidad de titular y autor de la tesis denominada "Modelado del desempeño de problemas NP-duros aplicando análisis causal para la herramienta VISTHAA" quien para todos los fines del presente documento se denominará **EL AUTOR Y/O TITULAR**, suscribo el presente documento de autorización de uso de derechos patrimoniales de autor a favor del Instituto Tecnológico de Ciudad Madero el cual se registrá por clausulas siguientes:

PRIMERA – AUTORIZACIÓN: EL AUTOR Y/O TITULAR, mediante el presente documento autoriza la utilización de los derechos patrimoniales de autor al Instituto Tecnológico de Ciudad Madero, de la tesis denominada "Modelado del desempeño de problemas NP-duros aplicando análisis causal para la herramienta VISTHAA", a través del Repositorio Institucional del Tecnológico Nacional de México (en lo sucesivo TecNM) y en el Repositorio Nacional, que puede ser consultado en la liga electrónica: (<https://www.repositorionacionalicti.mx/>).

SEGUNDA - OBJETO: Por medio del presente escrito, **EL AUTOR Y/O TITULAR** Autoriza al Instituto Tecnológico de Ciudad Madero, a través del Repositorio Institucional del Tecnológico Nacional de México (en lo sucesivo TecNM) y en el Repositorio Nacional para que de conformidad con la Ley Federal del Derecho de Autor y la Ley de la Propiedad Industrial, use los derechos del documento antes referido, con fines exclusivamente académicos.

TERCERA - TERRITORIO: Los derechos aquí Autorizados se dan sin limitación geográfica o territorial alguna.

CUARTA – ALCANCE: La presente autorización se da tanto para formato o soporte material, y se extiende a la utilización en medio óptico, magnético, electrónico, en red, mensajes de datos o similar conocido o por conocer, del ejemplar o número respectivo de la publicación.

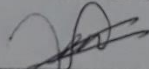
QUINTA – EXCLUSIVIDAD: La autorización de uso aquí establecida no implica exclusividad en favor del Instituto Tecnológico de Ciudad Madero. Por lo tanto **EL AUTOR Y/O TITULAR** en su carácter de autor de la obra objeto del presente documento se reserva el derecho de publicar directamente, u otorgar a cualquier tercero, autorizaciones de uso similares o en los mismos términos aquí acordados.

SEXTA - DERECHOS MORALES (Créditos y mención): La Autorización de los derechos antes mencionados no implica la cesión de los derechos morales sobre los mismos por cuanto en conformidad con lo establecido en los artículos 18, 19, 20, 21, 22 y 23 de la Ley Federal de Derechos de Autor, dada la cuenta que estos derechos son inalienables, imprescriptibles, irrenunciables e inembargables. Por lo tanto, los mencionados derechos seguirán radicados en cabeza de **EL AUTOR Y/O TITULAR**, y siempre deberá mencionarse su nombre cuando se utilice la obra.

SÉPTIMA - AUTORIA: **EL AUTOR Y/O TITULAR**, declara y ratifica que el material objeto de la presente y fue realizada por él (o ella) sin violar o usurpar derechos de Propiedad Intelectual de terceros.

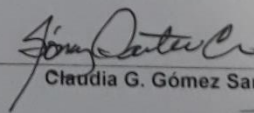
Ciudad Madero, Tamps. a 09 días del mes de Octubre de 2019.

Autor de la Tesis



Luis Rodolfo García Nieto

Asesor de la Tesis



Claudia G. Gómez Santillán

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

11 OCT 2019

RECIBIDO
DIVISIÓN DE ESTUDIOS DE POSGRADUADO E INVESTIGACIÓN



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN



"POR MI PATRIA Y POR MI BIEN"

TESIS
MODELADO DEL DESEMPEÑO DE PROBLEMAS NP-DUROS APLICANDO
ANÁLISIS CAUSAL PARA LA HERRAMIENTA VISTHAA

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

Presenta
I.S.C Luis Rodolfo Garcia Nieto
G10071394

Director de Tesis
Dra. Claudia Guadalupe Gómez Santillán

Co-director de Tesis
Dra. Laura Cruz Reyes

Cd. Madero, Tamaulipas

Octubre del 2019



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Madero
División de Estudios de Posgrado e Investigación

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cd. Madero, Tams., a **26 de Septiembre de 2019**

OFICIO No.: U5.224/19
ÁREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS.

LUIS RODOLFO GARCÍA NIETO
No. DE CONTROL G10071394
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:

"MODELADO DEL DESEMPEÑO DE PROBLEMAS NP-DUROS APLICANDO ANÁLISIS CAUSAL PARA LA HERRAMIENTA VISTHAA "

El Jurado está integrado por los siguientes catedráticos:

| | | |
|-----------------------|------|-----------------------------------|
| PRESIDENTE : | DR. | JOSÉ ANTONIO MARTÍNEZ FLORES |
| SECRETARIO: | DR. | NELSON RANGEL VALDEZ |
| VOCAL: | DRA. | CLAUDIA GUADALUPE GÓMEZ SANTILLÁN |
| SUPLENTE: | DR. | HÉCTOR JOAQUÍN FRAIRE HUACUJA |
| DIRECTORA DE TESIS : | DRA. | CLAUDIA GUADLUPE GÓMEZ SANTILLÁN |
| CO-DIRECTOR DE TESIS: | DR. | HÉCTOR JOAQUÍN FRAIRE HUACUJA |

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"*

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' 'mdcoa*



Av. 1° de Mayo y Sor Juana I. de la Cruz Col. Los Mangos, C.P. 89440, Cd. Madero, Tam.

Tel. 01 (833) 357 48 20, e-mail: direccion@itcm.edu.mx
www.tecnm.mx | www.cdmadero.tecnm.mx

CONTENIDO

Capítulo 1: Introducción

| | |
|-------------------------------------|----|
| 1.1 Motivaciones..... | 10 |
| 1.2 Antecedentes..... | 11 |
| 1.3 Estado del Arte..... | 11 |
| 1.4 Justificación..... | 12 |
| 1.5 Objetivos..... | 13 |
| 1.5.1 Objetivo General..... | 13 |
| 1.5.2 Objetivos Específicos..... | 13 |
| 1.6 Alcances y Limitaciones..... | 14 |
| 1.7 Organización del Documento..... | 14 |

Capítulo 2: Marco Teórico

| | |
|--------------------------------------------------------------------------------|----|
| 2.1 Problemas de Optimización..... | 15 |
| 2.1.1 Problema de la Mochila..... | 15 |
| 2.1.2 Problema de Empacado de Contenedores..... | 16 |
| 2.1.3 Problema de Empacado de Contenedores con Costos y Tamaños Variables..... | 16 |
| 2.1.4 Problema de Selección de Cartera de Proyectos..... | 17 |
| 2.2 Estrategias de Solución..... | 17 |
| 2.2.1 Métodos Exactos..... | 18 |
| 2.2.2 Algoritmos Heurísticos..... | 18 |
| 2.2.3 Algoritmos Metaheurísticos..... | 18 |
| 2.2.4 Algoritmos Genéticos..... | 19 |
| 2.3. Análisis Causal..... | 21 |
| 2.3.1 Modelado Causal..... | 22 |
| 2.3.2 Propiedades de un Modelo Causal..... | 22 |
| 2.3.3 Algoritmo PC..... | 23 |
| 2.4 Índices de Caracterización..... | 24 |
| 2.4.1 Índices del Problema..... | 25 |
| 2.4.2 Índices de la Muestra del Espacio de Soluciones..... | 29 |
| 2.4.3 Índices del Comportamiento del Algoritmo..... | 29 |
| 2.4.4 Índices de la Trayectoria del Algoritmo..... | 30 |
| 2.4.5 índices del Desempeño de Algoritmos Mono Objetivo..... | 31 |
| 2.4.6 Índices del Desempeño de Algoritmos Multi Objetivo..... | 32 |

| | |
|--------------------------------------------------------------------------------------------------|----|
| 2.5 Herramienta de Visualización de Análisis de Algoritmos Heurísticos (VisTHAA) | 34 |
| 2.5.1 Entrada de Datos y Preprocesamiento | 34 |
| 2.5.2 Caracterización de Instancias | 35 |
| 2.5.3 Visualización..... | 37 |
| 2.5.4 Módulo de Análisis Estadístico..... | 43 |
| Capítulo 3: Propuesta de Solución | |
| 3.1 Metodología del Módulo del Preprocesamiento | 47 |
| 3.2 Metodología del Módulo del Rediseño Causal | 49 |
| Capítulo 4: Experimentación y Análisis de Resultados | |
| 4.1 Ambiente Experimental..... | 62 |
| 4.2 Experimentación 1: Preprocesamiento de Instancias | 63 |
| 4.2.1 Resultados obtenidos del Preprocesamiento de Instancias..... | 65 |
| 4.3 Experimentación 2: Caracterización de la Estructura del Problema | 66 |
| 4.3.1 Resultados obtenidos del Proceso de Caracterización de la Estructura del Problema..... | 68 |
| 4.4 Experimentación 3: Caracterización del Comportamiento del Algoritmo..... | 69 |
| 4.4.1 Resultados obtenidos del Proceso de Caracterización del Comportamiento del Algoritmo | 70 |
| 4.5 Experimentación 4: Caracterización del Desempeño Algorítmico | 71 |
| 4.5.1 Resultados obtenidos del Proceso de Caracterización del Desempeño Algorítmico..... | 73 |
| 4.6 Experimentación 5: Rediseño Causal de Algoritmos Mono/Multi Objetivo | 74 |
| 4.6.1 Resultados obtenidos del Proceso de Caracterización | 75 |
| 4.6.2 Resultados obtenidos de la Identificación de las Regiones de Dominancia..... | 76 |
| 4.6.3 Resultados obtenidos de la Selección de Índices Significativos | 77 |
| 4.6.4 Resultados obtenidos del Análisis Causal..... | 79 |
| 4.6.5 Resultados obtenidos del Rediseño Causal | 80 |
| Capítulo 5: Conclusiones y Trabajos Futuros | |
| 5.1 Conclusiones..... | 86 |
| 5.2 Trabajos Futuros | 87 |
| Referencias | |

Índice de Figuras

| | |
|---------------------------------------------------------------------------------------------------------------|----|
| Figura 1 Clasificación del Proceso de Optimización. | 29 |
| Figura 2 Arquitectura Interna de VisTHAA. | 34 |
| Figura 3 Ruta de Acceso para la Carga de Instancias [Castillo11]. | 35 |
| Figura 4 Visualización de los Elementos de la Entrada de Datos [Castillo11]. | 35 |
| Figura 5 Módulo de Introducción de Nuevos Índices [Castillo11]. | 36 |
| Figura 6 Módulo de Generación de Matriz de Características [Castillo11]. | 37 |
| Figura 7 Ejemplo de Cálculo de un Índice Nuevo [Castillo11]. | 37 |
| Figura 8 Ruta del Módulo de Visualización de Instancias [Castillo11]. | 38 |
| Figura 9 Módulo de Visualización de Instancias de VisTHAA [Castillo11]. | 38 |
| Figura 10 Visualización Gráfica de los datos de Instancias Cargadas [Castillo11]. | 38 |
| Figura 11 Módulo de Visualización de Instancias en forma Ascendente de VisTHAA [Castillo11]. | 39 |
| Figura 12 Visualización Gráfica de los Datos de Instancias Cargadas en forma Ascendente [Castillo11]. | 39 |
| Figura 13 Ruta del Módulo de Visualización de Gráficas de Frecuencia [Castillo11]. | 40 |
| Figura 14 Módulo de Visualización de Gráficas de Frecuencia VisTHAA [Castillo11]. | 40 |
| Figura 15 Visualización Gráfica del Peso de los Elementos de la Instancia N1C1W1_A.BPP [Castillo11]. | 41 |
| Figura 16 Ruta del Módulo de Visualización de la Superficie de Aptitudes [Castillo11]. | 41 |
| Figura 17 Módulo de Visualización de la Superficie de Aptitudes [Castillo11]. | 42 |
| Figura 18 Visualización de la Superficie para la Instancia N1C1W1_A.BPP [Castillo11]. | 43 |
| Figura 19 Ruta del Módulo de Prueba de Wilcoxon [Castillo11]. | 43 |
| Figura 20 Módulo de Prueba Paramétrica de Wilcoxon [Castillo11]. | 44 |
| Figura 21 Ventana de Búsqueda de Archivo de Desempeño [Castillo11]. | 44 |
| Figura 22 Ventana de Resultados de la Prueba de Wilcoxon [Castillo11]. | 45 |
| Figura 23 Diagrama General de la Arquitectura de VisTHAA. | 46 |
| Figura 24 Ejemplo del Proceso de Almacenamiento de Instancias del Problema de la Mochila en VisTHAA. | 47 |
| Figura 25 Metodología del Rediseño Causal de Algoritmos Mono/Multi Objetivo. | 49 |
| Figura 26 Proceso de Caracterización Actual en VisTHAA. | 50 |
| Figura 27 Índice del Tamaño del Problema de la Variante 1. | 55 |
| Figura 28 Índice de la Muestra del Espacio de Soluciones de la Variante 1. | 55 |
| Figura 29 Índice de Comportamiento del Algoritmo de la Variante 1. | 56 |
| Figura 30 Índice de la Trayectoria del Algoritmo de la Variante 1. | 56 |
| Figura 31 Ejemplo del Mapeo de Índices para el Modelo Causal. | 57 |
| Figura 32 Proceso del Modelado Causal. | 58 |
| Figura 33 Elementos para la Creación del Modelo Causal en Tetrad. | 58 |
| Figura 34 Ejemplo del Modelo Causal del Algoritmo Genético en Tetrad. | 59 |
| Figura 35 Ventana Principal de VisTHAA. | 63 |
| Figura 36 Módulo Problem Selection en VisTHAA. | 64 |
| Figura 37 Ventana de Búsqueda del Archivo logbook. | 65 |
| Figura 38 Ventana de Instancias Cargadas del Problema de la Mochila en VisTHAA. | 65 |

| | |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 39 Ventana de Instancias Cargadas del Problema de Empacado de Contenedores con Pesos y Tamaños Variables en VisTHAA. | 66 |
| Figura 40 Ventana de Instancias Cargadas del Problema de Selección de Portafolio de Proyectos en VisTHAA. | 66 |
| Figura 41 Módulo de Attribute Matrix..... | 67 |
| Figura 42 Módulo Behavior Mono/Muti objective..... | 69 |
| Figura 43 Módulo Performance Mono Objective..... | 71 |
| Figura 44 Módulo Performance Multi Objective. | 72 |
| Figura 45 Módulo Algorithm Causal Redesign..... | 74 |
| Figura 46 Gráfica de la Capacidad Ocupada del Objeto de la Variante 6. | 78 |
| Figura 47 Gráfica de la Dispersión de los Pesos de la variante 6. | 78 |
| Figura 48 Gráfica de la Muestra del Espacio de Soluciones de la Variante 6. | 79 |
| Figura 49 Gráfica del Índice de la Trayectoria del Algoritmo de la Variante 6. | 79 |
| Figura 50 Modelo Causal obtenido para el Algoritmo NSGA-II..... | 80 |

Índice de Tablas

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabla 1 Trabajos Relacionados con la Caracterización y el Análisis Causal | 12 |
| Tabla 2 Elementos de Separación-D | 23 |
| Tabla 3 Índices del Tamaño del Problema. | 25 |
| Tabla 4 Índices de Relación entre los Pesos de los Objetos y el Tamaño del Contenedor..... | 25 |
| Tabla 5 Índices de Medidas de Tendencia Central de los Pesos..... | 26 |
| Tabla 6 Índices de Dispersión de Pesos: Rango, Error, Coeficiente de Variación, Varianza, Error Estándar..... | 26 |
| Tabla 7 Índices de la Posición de los Pesos..... | 27 |
| Tabla 8 Índices de la Forma de la Distribución de los Pesos. | 28 |
| Tabla 9 Índice de la Muestra del Espacio de Soluciones. | 29 |
| Tabla 10 Índices del Comportamiento del Algoritmo..... | 29 |
| Tabla 11 Índices de la Trayectoria del Algoritmo. | 30 |
| Tabla 12 Índices de la Caracterización del Desempeño de Algoritmos Mono Objetivo..... | 31 |
| Tabla 13 Índices de la Caracterización del Desempeño de Algoritmos Multi Objetivo. | 32 |
| Tabla 14 Listado de Métodos del Módulo de Preprocesamiento de VisTHAA. | 48 |
| Tabla 15 Ejemplos de Índices del Proceso de Caracterización..... | 51 |
| Tabla 16 Variantes del Algoritmo Metaheurístico Poblacional..... | 53 |
| Tabla 17 Representación Binaria de las Regiones de Dominancia del Algoritmo Solucionador. | 54 |
| Tabla 18 Ecuaciones del Modelo Causal del Algoritmo Genético. | 59 |
| Tabla 19 Algoritmo Metaheurístico Poblacional Rediseñado. | 60 |
| Tabla 20 Características del Equipo Experimental. | 62 |
| Tabla 21 Características del Conjunto de Problemas Analizados..... | 62 |
| Tabla 22 Características de Conjunto de Algoritmos Solucionadores. | 63 |
| Tabla 23 Matriz de Atributos del Problema de Empacado de Contenedores con Pesos y Tamaños Variables. | 68 |
| Tabla 24 Matriz de Atributos del Problema de Selección de Portafolio de Proyectos..... | 68 |
| Tabla 25 Resultados del Proceso de Caracterización del Comportamiento del VNS-VCBPP..... | 70 |
| Tabla 26 Resultados del Proceso de Caracterización del Comportamiento del Algoritmo NSGA-II. | 70 |
| Tabla 27 Resultados de la Caracterización del Desempeño del Algoritmo VNS-VCSBPP..... | 73 |
| Tabla 28 Resultados de la Caracterización del Desempeño del Algoritmo NSGA-II. | 73 |
| Tabla 29 Resultados obtenidos de los Índices del Proceso de Caracterización. | 75 |
| Tabla 30 Variantes del Algoritmo NSGA-II. | 76 |
| Tabla 31 Cantidad de Soluciones No Dominadas en el Frente de Pareto Óptimo por Variante..... | 77 |
| Tabla 32 Regiones de Dominancia de las Variantes del Algoritmo NSGA-II. | 77 |
| Tabla 33 Ecuaciones del Modelo Causal del Algoritmo NSGA-II..... | 80 |
| Tabla 34 Resultados obtenidos del Desempeño Algorítmico sin Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional. | 81 |
| Tabla 35 Resultados obtenidos del Porcentaje de Error sin Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional. | 81 |
| Tabla 36 Resultados obtenidos del Desempeño Algorítmico con Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional. | 81 |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabla 37 Resultados obtenidos del Porcentaje de Error con Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional. | 82 |
| Tabla 38 Resultados obtenidos del Porcentaje de Mejora del Desempeño del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional. | 82 |
| Tabla 39 Resultados obtenidos de la Proporción de Soluciones No Dominadas sin la aplicación del Rediseño Causal del Conjunto de Variantes del Algoritmo NSGA-II. | 83 |
| Tabla 40 Resultados obtenidos de la Proporción de Soluciones No Dominadas después de la aplicación del Rediseño Causal del Conjunto de Variantes del Algoritmo NSGA-II. | 83 |
| Tabla 41 Porcentaje de mejora del Índice de Proporción de Soluciones No Dominadas del Conjunto de Variantes del Algoritmo NSGA-II. | 83 |
| Tabla 42 Resultados obtenidos de la Cantidad de Soluciones pertenecientes del Frente de Pareto Optimo sin la aplicación del Rediseño Causal. | 84 |
| Tabla 43 Resultados obtenidos de la Cantidad de Soluciones pertenecientes del Frente de Pareto Optimo después de la aplicación del Rediseño Causal. | 84 |
| Tabla 44 Porcentaje de mejora del Número de Soluciones pertenecientes al Frente de Pareto Optimo del Conjunto de Variantes del Algoritmo NSGA-II. | 84 |

Capítulo 1: Introducción

En esta sección se presenta un panorama general de la tesis, iniciando con la descripción de los motivos que originaron esta investigación, los antecedentes y el estado del arte que brindan información para la realización de la misma. Enseguida se explican los objetivos que se plantearon a cumplir, así también se describen los alcances y limitaciones del este trabajo de investigación. Y por último se presenta una breve descripción del contenido de cada capítulo de la tesis.

1.1 Motivaciones

Una buena alternativa para la solución de instancias pertenecientes a los problemas de optimización combinatoria es, a través de algoritmos metaheurísticos [Garey79, Merz99]. Actualmente los investigadores desean seleccionar el mejor algoritmo para la solución de problemas reales de la vida cotidiana, usualmente no hay un algoritmo que supere a otros algoritmos para todas las instancias a analizar [Wolpert97]. El problema de selección de algoritmos metaheurísticos es un problema abierto de la computación [Anderson03].

Una alternativa para la identificación y selección del mejor algoritmo es a través del proceso de caracterización, el cual permite a los investigadores el caracterizar, ya sea la estructura, comportamiento o desempeño del algoritmo, proporcionando fundamentos para la selección del mejor. Algunos trabajos que incluyen la caracterización son [Quiroz09, Quiroz14].

Actualmente hay áreas de la computación que se encargan del análisis del desempeño de algoritmos y algunos autores proponen el mejoramiento a través de una metodología causal [Perez07, Landero08].

En la literatura existe poca documentación de herramientas que cuentan con funciones que permitan la caracterización de algoritmos e instancias a través de índices y de herramientas que proporcionen un análisis causal como los es Tetrad y Hugin.

En este trabajo se propone la implementación de módulos que permitan la caracterización y el análisis causal a algoritmos que dan solución a problemas NP-Duros pertenecientes a la familia de contenedores. Para su cumplimiento se abordaron dos procesos, los cuales son: el preprocesamiento de instancias pertenecientes a problemas NP-Duros mono/multi objetivo, que forman parte de la familia de contenedores y el rediseño de algoritmos, que dan solución a los tipos de problemas mencionados anteriormente, a través de una metodología causal que incorpora la caracterización de la estructura del problema, espacio de soluciones, comportamiento, trayectoria y desempeño algorítmico, con el objetivo de mejorar su rendimiento a través del rediseño causal.

Dichos procesos se llevaron a cabo en la herramienta de VisTHAA, con el objetivo de brindar a los investigadores un mejor diseño de los algoritmos, a través de módulos que incorporan el análisis causal e índices estadísticos, los cuales apoyaran a los investigadores a dar explicaciones del comportamiento de los mismos.

Para el cumplimiento de éste trabajo, primero se rediseño el módulo de preprocesamiento, con el objetivo de procesar instancias multi objetivo y además se incorporaron métodos que permiten el procesamiento de más problemas mono objetivo. En el capítulo 3 se explica la metodología aplicada del rediseño del módulo de preprocesamiento de la herramienta VisTHAA.

Además se incorporaron un conjunto de módulos, permitiendo al investigador la caracterización de la estructura del problema, comportamiento y desempeño algorítmico, las cuales son subprocesos

independientes de la metodología del rediseño causal. Y un módulo que permite la mejora del desempeño a través del modelado causal.

1.2 Antecedentes

En esta sección se explican el conjunto de áreas de conocimiento que brindarán información para la realización del trabajo final de ésta tesis. Dichas áreas de investigación en las cuales han trabajado algunos investigadores son:

1. Caracterización de algoritmos, instancias y resultados.
2. Análisis causal enfocado en el comportamiento del algoritmo a través de la aplicación del rediseño causal para su mejoría.
3. Visualización de los incisos 1) y 2) permitiendo al investigador mostrar de forma gráfica conjunto de índices calculados.

Algunos trabajos en el área de las ciencias de la computación, han incursionado en la aplicación de modelos causales para el análisis de algoritmos de solución para problemas NP-Duros. Las técnicas usadas por las tecnologías de información se encuentran en un proceso de evolución, destacando el uso de algoritmos inteligentes para resolver problemas reales complejos. Con el propósito de desarrollar mejores estrategias, diversos investigadores han intentado analizar y explicar el comportamiento los algoritmos NP-Duros con diferentes instancias de un problema, partiendo del análisis experimental, analizando estadísticamente los datos obtenidos.

El trabajo de [Perez07] propone el análisis de algoritmos metaheurísticos en problemas de distribución de objetos a través de una metodología causal, permitiendo la identificación de las relaciones existentes entre factores que influyen el desempeño de un algoritmo, la explicación del desempeño algorítmico y la interpretación del porque un algoritmo se comporta mejor sobre un conjunto particular de instancias.

El trabajo de [Lander08] propone una metodología que permite la construcción de modelos causales a través de un conjunto de índices de la estructura del problema, espacio de soluciones, comportamiento, trayectoria y desempeño algorítmico, con el objetivo de explicar formalmente porqué un algoritmo heurístico de aproximación es el más adecuado para resolver un subconjunto de instancias del problema Bin-Packing.

Por último, el trabajo de [Castillo11] propone una herramienta de visualización que permite mostrar de manera gráfica índices, con el objetivo de mejorar el desempeño de los algoritmos. Dicha herramienta se le denominó VisTHAA (Visualization Tool for Heuristic Algorithms Analysis), que permite al investigador la caracterización del desempeño de los algoritmos. Esta herramienta cuenta con 3 módulos: el módulo de entrada, el módulo de preprocesamiento de datos y el módulo de caracterización.

1.3 Estado del Arte

En esta sección se hace mención de algunos trabajos previos, que contribuyeron a la realización de este proyecto, específicamente aquellos trabajos que estén relacionados con el proceso de caracterización de problemas pertenecientes a la familia de contenedores, del desempeño de algoritmos, y el análisis causal. En la Tabla 1 se muestra algunos de los trabajos relacionados con la caracterización y el análisis causal.

Tabla 1 Trabajos Relacionados con la Caracterización y el Análisis Causal

| Investigador | Tema del trabajo de investigación |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------|
| [Perez07] | “Modelado causal del desempeño de algoritmo metaheurísticos en problemas de distribución de objetos” |
| [Landro08] | “Metodología para la explicación causal del comportamiento y desempeño de algoritmos” |
| [Quiroz09] | “Caracterización de factores de desempeño de algoritmos de solución de BPP” |
| [Quiroz14] | “Caracterización del proceso de optimización de algoritmos heurísticos aplicados del problema de empaçado de objetos en contenedores” |
| [Mexicano15] | “A través de la caracterización de instancias difíciles del problema de empaçado de contenedores” |
| [Martinez17] | “Adaptación de un algoritmo de agrupación mono objetivo a multi objetivo usando caracterización del desempeño” |

En el trabajo de [Pérez07] se analizó el desempeño de algoritmo metaheurístico en problemas de distribución de objetos utilizando análisis causal. En su trabajo se identificaron y seleccionaron factores relacionados con el desempeño de algoritmo búsqueda Tabú, para construcción de modelos causales, para la explicación del desempeño algorítmico en un conjunto particular de instancias. El trabajo de [Landro08] propone una metodología para la mejora del desempeño de un conjunto de algoritmos a través de una metodología de generación de modelos causales.

En el trabajo de [Quiroz09] se identificaron estrategias heurísticas para la incorporación de algoritmos metaheurísticos para la mejora del desempeño; además se propuso una metodología para la generación de modelos causales, el cual permite la identificación y selección de factores relacionados con el desempeño del algoritmo metaheurístico, y además permite encontrar las relaciones entre los factores que influyen el desempeño para dar explicación formal y poderlo mejorar.

En el trabajo de [Quiroz14] se identificaron propiedades que definen una instancia difícil de resolver, además de caracterizar el comportamiento de algoritmo e identificar estrategias y condiciones que permitieron alcanzar un alto desempeño a través del modelado de las relaciones entre las características del problema y de comportamiento algorítmico.

[Mexicano15] muestra en su trabajo un seguimiento de los algoritmos metaheurísticos a través de la identificación de características comunes en instancias del BPP que son difíciles de resolver mediante métodos convencionales.

En el trabajo de [Martínez17] se identificaron métodos de análisis experimental que explican el proceso de optimización de un algoritmo heurístico y mediciones representativas que permitieron caracterizar el proceso de optimización de un algoritmo heurístico multi objetivo.

El proceso del rediseño causal es un problema desafiante, y aún existe un camino por correr para la generación de una metodología que siempre mejore el desempeño algorítmico en todos los casos, o que el desempeño obtenido sea el mejor reportado en la literatura. Se espera que la aplicación del rediseño causal permita mejorar el desempeño de algoritmos que dan solución a un conjunto de instancias de problemas pertenecientes a la familia de empaçado de contenedores.

1.4 Justificación

El principal objetivo del análisis de la información, es el descubrimiento de conocimiento útil ya que este debe ser transportable e independiente del contexto, permitiendo así su generalización y aplicación a diversas problemáticas. Para ello se utilizan algunas estrategias entre ellas el análisis causal.

El trabajo de [Perez07] menciona que el proceso de crear un modelo causal no es trivial, debido a que es un problema NP-Completo, lo cual implica una alta complejidad computacional.

La construcción de un modelo causal consiste principalmente en la representación del conocimiento obtenido a través de modelos que permiten mostrar dependencias de relaciones causa y efecto de un conjunto de datos. A grandes volúmenes de datos, el costo de la creación de modelos causales es exponencial, por este motivo es importante conocer que factores afectan el desempeño de dichos algoritmos, muchos investigadores han utilizado la caracterización como una herramienta que además de explicar el comportamiento de los algoritmos, permite también obtener un conocimiento a detalle de cada una de las fases del proceso de solución y entender su conducta.

El conocimiento obtenido de las relaciones causales entre las características de los datos es posible utilizarlo para crear guías [Glymour99]. Trabajos previos en las Ciencias de la Computación han incursionado en la aplicación de modelos causales para el análisis de algoritmos de solución para problemas complejos, ya que aportarían elementos para un mayor entendimiento del comportamiento y de los algoritmos a utilizar, los cuales se muestran en la Tabla 1.

Actualmente la herramienta VisTHAA no cuenta con métodos que permitan el proceso de rediseño causal, para algoritmos que dan solución a problemas mono/multi objetivos pertenecientes a la familia de contenedores.

En este trabajo se plantea el rediseño del módulo de preprocesamiento de instancias, para la aceptación de problemas NP-Duros multi objetivo y de más problemas mono objetivo, pertenecientes a la familia de contenedores, y la implementación de un módulo que permita el rediseño de algoritmos mono/multi objetivo a través de la caracterización de la estructura del problema, comportamiento, trayectoria y desempeño algorítmico, permitiendo a los investigadores, dar una explicación formal del comportamiento de algoritmos que dan solución a problemas NP-Duros.

1.5 Objetivos

A continuación, se presentan el objetivo general y los objetivos específicos planteados para éste trabajo de investigación.

1.5.1 Objetivo General

Mejorar el rendimiento de algoritmos que dan solución a problemas NP-Duros de la familia de contenedores, aplicando caracterización y análisis estadístico en VISTHAA.

1.5.2 Objetivos Específicos

1. Analizar VISTHAA para proponer un nuevo módulo que permita hacer el rediseño de un algoritmo.
2. Rediseño del módulo de preprocesamiento de instancias de la Herramienta VISTHAA para soportar problemas multi objetivo.
3. Analizar estrategias de los algoritmos causales para el estudio de problemas NP-Duros multi objetivo.
4. Diseño e implementación de las estrategias seleccionadas en los puntos mencionados anteriormente.
5. Validar resultados obtenidos de mejoría en la calidad de las soluciones de los algoritmos.

1.6 Alcances y Limitaciones

En esta sección se explican los alcances y limitaciones de este trabajo de investigación

1. Se trabajará con un problema multi objetivo del tipo de mochila o cartera de proyectos o bin packing, etc.
2. Se modificará el módulo de preprocesamiento para generalizar las estructuras de datos y aceptar problemas multi objetivo.
3. Se añadirá por lo menos una prueba estadística que valide el rendimiento del algoritmo en el módulo de “caracterización del rendimiento final” para problemas multi objetivo.
4. Se añadirá una opción en el módulo de caracterización estructural del problema para incluir estrategias de análisis causal.

1.7 Organización del Documento

En esta sección se presenta una breve descripción de los capítulos que conforman esta tesis.

1. El Capítulo 2 se enfoca en los fundamentos y conceptos teóricos que darán facilidad al entendimiento de esta tesis, entre ellos, los problemas de optimización a estudiar, estrategias de solución de los algoritmos de optimización a abordar, conceptos y métodos del análisis causal, índices de caracterización y de la herramienta VisTHAA.
2. En el Capítulo 3 se explica la metodología utilizada para el cumplimiento de los objetivos específicos mencionados en la sección 1.4.2, entre ellos están la metodología usada para el preprocesamiento de instancias mono objetivo y multi objetivo, la caracterización de instancias, desempeño de los algoritmos solucionadores y el rediseño causal de algoritmos mono objetivo y multi objetivo.
3. En el Capítulo 4 se muestran los resultados obtenidos de las experimentaciones realizadas de las metodologías propuestas para los puntos citados en el Capítulo 3, y la implementación e incorporación de dichos puntos mencionados a la herramienta VisTHAA.
4. En el Capítulo 5 se reportan las conclusiones obtenidas de la investigación realizada, y de igual manera se presentan las sugerencias para trabajos futuros.

Capítulo 2: Marco Teórico

En este capítulo se presenta una revisión de conceptos y enfoques esenciales relacionados con el desarrollo de este trabajo de investigación. Se describen los problemas de optimización a abordar, estrategias de solución para problemas de optimización mono objetivos y multi objetivo, técnicas causales que permitirán realizar un rediseño causal a los algoritmos a usar, se mencionarán métricas que permitirán medir el desempeño de los algoritmos mono objetivo y multi objetivo a utilizar, y se dará introducción de la herramienta VisTHAA la cual contendrá nuevos módulos a proponer.

2.1 Problemas de Optimización

Un problema de optimización consiste en encontrar la mejor solución posible dentro de un espacio de soluciones, en donde la mejor solución es considerada con respecto a un criterio máximo o mínimo. En este trabajo de investigación se abordarán los siguientes casos de estudio:

1. El problema de la mochila mono objetivo (Knapsack problem)
2. El problema de empaqueo de contenedores (Bin packing problem)
3. El problema de empaqueo de contenedores con pesos y tamaños variables (Variable cost and sized bin packing problem)
4. El problema de selección de cartera de proyectos multi objetivo

2.1.1 Problema de la Mochila

El problema de la mochila es uno de los problemas clásicos de optimización combinatoria, ya que es clasificado como un problema NP-Duro [Cruz04]. Es un problema de optimización combinatoria que juega un rol importante en la teoría de la computación, ya que tiene como meta la búsqueda de la mejor solución entre un conjunto finito de posibles soluciones a un problema.

El objetivo del problema es seleccionar la mayor cantidad de objetos, que se llevarán en la mochila, maximizando su beneficio sin sobrepasar la capacidad de la misma [Martello90].

$$\text{maximizar } Z = \sum_{j=1}^N C_j X_j \quad (1)$$

$$\text{s. a } \sum_{j=1}^N x_j p_j \leq P \quad (2)$$

$$X_j \in \{0,1\}, \quad j \in N = \{1,2, \dots, n\} \quad (3)$$

Donde:

- La función objetivo (1) maximiza el beneficio total Z del beneficio de los objetos C_j asignados a la mochila.
- La restricción (2) asegura que el peso total p_j de los objetos x_j no sobrepase la capacidad de P .
- La restricción (3) exige que las variables de decisión sean binarias, si x_j es tal que $x_j = 0$ si el objeto j no está en la mochila, $x_j = 1$ en caso contrario.

2.1.2 Problema de Empacado de Contenedores

El problema de empaado de objetos en contenedores en una dimensión, consiste en almacenar objetos de diferentes tamaños, o pesos en el menor número de contenedores de tamaño fijo. Este es un problema clásico de optimización combinatoria NP-Duro, considerado intratable ya que demanda una gran cantidad de recursos para su solución [Basse98, Alvarez06].

$$\text{minimizar } Z = \bigcup_{j=1}^m B_j \quad (1)$$

$$\text{s. a. } \sum_{\forall i \in B_j} w_i \leq c \quad 1 \leq j \leq m \quad i \in N = \{1, \dots, n\} \quad (2)$$

Donde:

- La función objetivo (1) consiste en asignar cada objeto a un contenedor de tal forma que la suma de los pesos de los objetos en cada contenedor no exceda c y el número de contenedores m utilizado sea el mínimo [Martello90]. Es decir, se busca encontrar el menor número de subconjuntos m para una partición del conjunto N .
- Y la restricción (2) indica que el peso del objeto i (w_i) sea menor que la capacidad de cada contenedor C y N es el número de objetos a distribuir en el contenedor.

2.1.3 Problema de Empacado de Contenedores con Costos y Tamaños Variables

El problema de empaado de contenedores con costos y tamaños variables en sus siglas en inglés (VCSBPP), tiene como objetivo la minimización del costo total de los contenedores seleccionados. Este problema puede ser formalmente descrito de la siguiente manera [Crainic11].

Sea X ($|X| < \infty$) la representación del conjunto de objetos que serán cargados, cada objeto $i \in X$ tiene un volumen v_i . Sea T ($|T| < \infty$) el conjunto de contenedores disponibles V_j y c_j representan el volumen y el costo del contenedor $j \in T$ respectivamente. Sin ninguna pérdida de la generalidad, se asume que los volúmenes y los costos asociados a los contenedores y a los objetos son enteros

Sean las variables de selección de contenedores $y = (y_1, y_2, \dots, y_{|T|})$ en donde $y_j = 1$, si el contenedor j es seleccionado y $y_j = 0$ en caso contrario. Sean las variables de asignación objeto-contenedor $x_{ij}, \forall i \in X, \forall j \in T$, en donde $x_{ij} = 1$, si el objeto i es cargado en el contenedor j y $x_{ij} = 0$ en caso contrario. El VCSBPP puede ser modelado de la siguiente manera:

$$\min z = \sum_{j \in T} c_j * y_j \quad (1)$$

s.a:

$$\sum_{j \in \tau} x_{ij} = 1, \quad \forall i \in X \quad (2)$$

$$\sum_{i \in X} v_i x_{ij} \leq V_j y_j, \quad \forall j \in \tau \quad (3)$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in X, \quad \forall j \in \tau \quad (4)$$

$$y_j \in \{0,1\}, \quad \forall j \in \tau \quad (5)$$

En donde:

- La función objetivo (1) minimiza el costo total de los contenedores seleccionados.
- El conjunto de restricciones (2) se asegura que cada objeto $i \in X$ este empacado exactamente en un contenedor.
- El conjunto de restricciones (3) se asegura que el volumen total de los objetos empacados en el contenedor $j \in \tau$ o exceda su volumen V_j .
- Las restricciones (4) y (5) exigen que las variables de decisión sean enteras.

2.1.4 Problema de Selección de Cartera de Proyectos

La correcta selección de proyectos para integrar una cartera de proyectos es uno de los problemas más importantes de decisión, tanto para instituciones públicas como privadas.

Los principales modelos económicos y matemáticos para el problema de cartera de proyectos, suponen que se tiene un conjunto definido N de proyectos, cada proyecto perfectamente caracterizado con costos e ingresos, de los cuales la distribución en el tiempo es conocido.

En caso de riesgos, el "Tomador de Decisiones" (DM) debe conocer la probabilidad de distribución de los beneficios. El DM es una persona o un grupo de personas a cargo de seleccionar, a su juicio y dada su experiencia, las mejores soluciones. [Fernandez01]. El modelo matemático del problema se expresa de la siguiente manera:

$$\text{maximizar } Z = \sum_{j=1}^N c_j x_j \quad (1)$$

Donde:

- La función objetivo (1) maximiza el beneficio total asociado Z a una cierta cartera de proyectos, c_j es el beneficio asociado con el proyecto j y $x_j = 1$ indica si el proyecto j forma parte de la cartera y $x_j = 0$ en caso contrario.
- Sea $x(j)$ la función indicadora del conjunto de los proyectos finalizados donde $x(j) = 1$ si el proyecto j recibe el financiamiento solicitado y $x(j) = 0$ en caso contrario.

2.2 Estrategias de Solución

En este apartado se presentarán algunas de las estrategias a utilizar para dar solución a los problemas de optimización, dichos algoritmos se incorporarán a la herramienta para que crezca su robustez y funcionalidad.

2.2.1 Métodos Exactos

Este tipo de soluciones exploran todo el espacio de soluciones de un problema, y siempre encuentra el mejor camino posible. Algunos de esos métodos son el método Band-and-Price [Baldi12], algoritmos voraces, algoritmos de dividir y vencerás, back tracking [Duarte07] entre otros.

Los métodos exactos pueden resolver los problemas pertenecientes a la clase P (polinomio), debido a que son resueltos por algoritmos con complejidad polinomial en un tiempo razonable [Gomez14], esto es porque estos tipos de problemas tienen la característica que pueden ser resueltos por un algoritmo determinista [Garey79]. Sin embargo, hay una clase de problemas para los cuales los algoritmos deterministas exactos no tienen una convergencia en tiempo polinomial. Este tipo de problemas pertenecen a la clase NP (problema del polinomio-tiempo no determinista) que son aquellos problemas que pueden ser transformados (reducidos) a un problema de decisión que es resuelto por un algoritmo no determinista en un tiempo polinomial [Gomez14], para ellos, es necesario aplicar otros tipos de algoritmos. Porque a pesar de que un método exacto va a encontrar la mejor solución, tomaría tanto tiempo encontrarlo, ya que lo hace completamente irrelevante.

2.2.2 Algoritmos Heurísticos

Muchos problemas de optimización no se pueden resolver mediante métodos exactos. Esto se debe a la gran cantidad de tiempo que se tarda en resolver un problema de gran complejidad. Para este tipo de problema, los métodos conocidos como heurísticas se han generado desde la década de 1960 [Polya45]. Estos se caracterizan por su capacidad para encontrar soluciones de buena calidad (incluso si no son las mejores) en un tiempo razonable.

2.2.2.1 Algoritmo de Adaptación del Mejor Ajuste Decreciente

El algoritmo de adaptación de mejor ajuste decreciente es uno de los algoritmos más populares para el problema de empaqueo de contenedores [Kang03], basado en el funcionamiento de colocar cada objeto en el mejor contenedor disponible. [Creinic11] propone en su trabajo una versión modificada del algoritmo para la solución del problema del VCSBPP, a través de la ordenación de los objetos de acuerdo con el orden no-creciente de sus volúmenes y después colocarlos de manera secuencial.

Este algoritmo funciona de la siguiente manera: cada objeto se intenta colocar en el mejor contenedor seleccionado, maximizando el espacio libre del mismo, si el objeto ya no puede ser colocado en un contenedor, se escoge uno nuevo y se coloca el objeto.

2.2.3 Algoritmos Metaheurísticos

Varios algoritmos heurísticos tienen la problemática del estancamiento en óptimos locales, esto es debido a que no tienen un mecanismo para continuar con la búsqueda de la mejor solución. Para resolver dicha problemática existen otro tipo de algoritmos llamado-metaheurísticos que tienen procedimientos para evitar que los algoritmos heurísticos no queden atrapados en óptimos locales [Duarte07].

2.2.3.1 Algoritmo de Búsqueda de Vecindad Variable del Problema de Empacado de Contenedores con Pesos y Tamaños Variables

La búsqueda de vecindad variable (VNS) es una metaheurística que tiene como objetivo el evitar quedar atrapado en óptimos locales, utilizando el manejo de estructuras de vecindad [Mladenovic97]. Su principal funcionamiento consiste en la exportación de vecindades lejanas de una solución actual, y moverse de una solución a hacia una solución b , si, y solo si la nueva solución es mejor que la actual.

El trabajo de [Duarte07] menciona que la metaheurística VNS se basa en tres puntos:

1. Un óptimo local con respecto a una vecindad $N_i(x)$ no tiene por qué serlo con respecto a otra vecindad $N_j(x)$.
2. El óptimo global es un óptimo local con respecto a todas las posibles estructuras de vecindad.
3. Para muchos problemas, los óptimos locales con respecto a una o varias estructuras de vecindad, están relativamente próximos.

La metaheurística funciona de la siguiente manera:

Primero se genera una solución inicial utilizando la heurística A-BFD con el objetivo de comenzar con una buena solución cercana a la solución óptima, después en cada iteración se escoge una de las dos estructuras disponibles utilizando una selección por ruleta, en donde siempre serán favorecidas las estructuras de vecindad que resulten más exitosas. A continuación, se muestra en el Algoritmo 1 el proceso de búsqueda de vecindad variable descrito en el trabajo de [Padilla18].

Algoritmo 1 Proceso de Búsqueda de Vecindad Variable [Padilla18].

- 1) Entrada: una instancia del problema, estructuras de vecindades N_k , $k \in \{1,2\}$
- 2) $x \leftarrow \text{GenerarSolucionInicial}()$
- 3) $c_1 = 1, c_2 = 1$
- 4) Mientras no se cumpla el criterio de paro hacer
- 5) $k \leftarrow \text{EscogerVecindad}(c_1, c_2)$
- 6) $x' \leftarrow \text{Sacudir}(x, k)$
- 7) $x'' \leftarrow \text{BusquedaLocal}()$
- 8) Si x'' tiene un valor de costo menor que x entonces
- 9) $x \leftarrow x''$
- 10) $c_k \leftarrow c_k + 1$
- 11) Fin si
- 12) Fin mientras
- 13) Salida: Mejor solución encontrada

2.2.4 Algoritmos Genéticos

Estos tipos de métodos, se basan en la recombinación de soluciones candidatas de una población, que evoluciona por medio de mecanismos genéticos como lo es la selección, la cruce y la mutación de los individuos de una población.

Los algoritmos genéticos (Genetic Algorithm - GA) fueron presentados por J. Holland [Holland95].

El concepto de recombinación de soluciones supone una de las aportaciones fundamentales de GA. Por otro lado, es también relevante la diferencia explícita entre la representación del problema (denominada genotipo), que habitualmente viene determinada por cadenas de bits conocidas como cromosomas, y las variables del problema en sí (denominadas fenotipo). GA representa una metaheurística poblacional sencilla e intuitiva que, muy probablemente, sea la más utilizada. El principio de operación es el siguiente:

“GA se basa en una población de soluciones candidatas, que evoluciona por medio de los mecanismos genéticos neodarwinistas de selección, cruce y mutación”.

La nomenclatura utilizada en GA está muy relacionada con su inspiración biológica. En este sentido, es usual denominar “población de individuos” al conjunto de soluciones, de forma que cada individuo

se corresponde con una solución. La solución en sí (variables del problema) establece el fenotipo del individuo, y su representación el genotipo. Esta representación recibe el nombre de cromosoma y cada uno suele estar compuesto por unidades discretas llamadas genes. GA es un tipo de EA, y heredan sus características algorítmicas. Por esta razón, se ha omitido de esta sección el algoritmo que describe su pseudocódigo. Sin embargo, GA presenta una serie de particularidades que lo hace singular y con entidad propia como metaheurística. Los elementos que deben identificarse para resolver un problema con un algoritmo genético son:

Población inicial: Suele estar formada por una generación aleatoria de soluciones al problema dado.

Representación: Constituye una correspondencia entre las soluciones factibles (fenotipo) y la codificación de las variables (genotipo). Originalmente, las representaciones eran cadenas binarias. Sin embargo, actualmente se han utilizado otras representaciones en problemas discretos, de permutaciones y binarios.

Función de evaluación: Determina la calidad de los individuos de la población. Habitualmente, es una función monótona creciente que asigna un valor mayor cuanto mejor sea el individuo.

Operadores genéticos: Son métodos probabilísticos que obtienen nuevos individuos. Suelen ser dependientes de la representación. Habitualmente, se utilizan los siguientes operadores:

1. **Cruza:** Consiste en la sustitución de un conjunto de genes de un padre por los genes correspondientes del otro padre para generar un nuevo individuo hijo heredando características de sus padres. Algunas técnicas básicas de cruce son las siguientes:

1. Cruza de un punto
2. Cruza de dos puntos
3. Cruza uniforme

Además, existen estrategias de cruce a base de permutaciones, las cuales se mencionan a continuación:

1. Order Crossover (OX)
2. Partially Mapped Crossover (PMX)
3. Position-based Crossover
4. Order-based Crossover
5. Cycle Crossover (CX)

2. **Mutación:** Es un operador secundario en los algoritmos genéticos, el cual tiene un uso menos frecuente que la cruce ya que utiliza porcentajes bajos de probabilidad, para que ocurra (se recomienda entre 0.001 y 0.01 para representación binaria).

El proceso consiste en cambiar de manera drástica un cierto conjunto de valores del cromosoma de un individuo, con el objetivo de obtener un cambio repentino en varios miembros de la población, para de esta manera ver si dichos cambios provocan una mejora en la población a corto y mediano plazo.

Las estrategias de mutación para permutaciones más conocidas son:

1. Mutación por inserción
2. Mutación por desplazamiento
3. Mutación por intercambio recíproco

4. Mutación heurística

3. **Selección:** Es un mecanismo que permite elegir con mayor probabilidad a los individuos que presenten un valor más elevado de la función de evaluación. Existen diferentes estrategias de selección, a continuación se muestran algunas de las más conocidas:

1. Selección proporcional
 - a. Ruleta
 - b. Sobrante estocástico
 - c. Universal estocástica
 - d. Muestreo determinístico
 - e. Escalamiento sigma
 - f. Selección por jerarquías
 - g. Selección de Boltzmann
2. Selección mediante torneo
3. Selección de estado uniforme

2.2.4.1 Algoritmo Genético de Ordenamiento por No Dominancia

Es un algoritmo genético popular basado en la no dominancia para la optimización multi objetivo [Srinivas94], el cual tiene incorporado un algoritmo de ordenamiento y elitismo [Deb02]. En el Algoritmo 2 se muestra el proceso de solución del NSGA-II.

Algoritmo 2 Proceso del NSGA-II.

1. Procedimiento NSGA-II ($N', g, f_k(x_k)$) $\rightarrow -N'$ miembros evolucionados de g generaciones para resolver $f_k(x)$
2. **Inicializar** población \mathbb{P}'
3. **Generar** población aleatoria tamaño N'
4. **Evaluar** la función objetivo
5. **Asignar** ranqueo (nivel) basado en el ordenamiento de dominancia de Pareto
6. **Generar** población de hijos
7. **Selección** de torneo binario
8. **Recombinación y mutación**
9. For $i=1$ to g hacer
10. For cada padre e hijo en la población hacer
11. **Asignar** ranqueo (nivel) basado en el ordenamiento de Pareto
12. **Generar** conjunto de vectores no dominados $\rightarrow PF_{known}$
13. **Agregar** las soluciones a la siguiente generación iniciando del primer frente hasta los N' individuos encontrados, determinando la distancia crowding entre puntos de cada frente
14. Fin For
15. **Crear** siguiente generación
16. **Selección** torneo binario
17. **Recombinación y mutación**
18. Fin for
19. **Fin** Procedimiento

2.3. Análisis Causal

La causalidad es un término que indica cómo el mundo responde a una intervención. En la vida cotidiana existen muchas situaciones en las que podemos observar la causalidad. El primer paso para identificar la causalidad es la asociación de eventos. Por ejemplo, hacer la aseveración siguiente: ver

demasiada televisión provoca un aumento de peso, relaciona ver televisión en exceso con aumentar de peso.

Este es el primer paso para identificar causalidad, el siguiente es intervenir. Siguiendo el ejemplo, hacer un experimento en el que un grupo de individuos vea televisión en exceso y registrar su peso diariamente podría llevarnos a la conclusión de que efectivamente aumentaron de peso.

En el ejemplo, la causalidad puede indicarse como el proceso de identificar la relación directa que existe entre dos eventos. Si la relación causal es invariante a las intervenciones o cambios, permite la predicción de acciones. Es por esto por lo que un modelo causal es una extensión de los modelos de predicción, y que proporciona un refinamiento causal a las reglas basadas en semántica lógica [Perez07].

2.3.1 Modelado Causal

Un modelo causal, es una representación generalizada del conocimiento obtenido al encontrar dependencias que impliquen relaciones de causa y efecto. La causalidad requiere identificar la relación directa existente entre eventos o variables. Las variables que intervienen en el modelo son de naturaleza aleatoria, y algunas pueden tener relación causal con otras. En la práctica las variables se dividen en dos conjuntos, las variables exógenas, cuyos valores son determinados por factores fuera del modelo y las endógenas, que tienen valores descritos por un modelo de ecuaciones estructurales.

Las relaciones causales son transitivas, irreflexivas y antisimétricas. Esto quiere decir que: 1) si A es causa de B y B es causa de C , entonces A es además causa de C , 2) un evento A no puede causarse a sí mismo, y 3) si A es causa de B entonces B no es causa de A .

La representación de un sistema causal se hace a través un grafo acíclico dirigido (DAG), que muestra las influencias causales entre las variables del sistema y ayuda a estimar los efectos totales y parciales que resultan de la manipulación de una variable. Un modelo causal puede ser definido como una red Bayesiana causal [Spirtes01], el cual está compuesta por un modelo causal, un conjunto de variables, un grafo dirigido acíclico y una función de probabilidad condicional.

2.3.2 Propiedades de un Modelo Causal

Los grafos dirigidos acíclicos son interpretados causalmente cuando éstos poseen ciertas propiedades que hacen la conexión entre la causalidad y la probabilidad [Glymour99, Spirtes01].

- G un grafo que representa una estructura causal $G = (V, E)$.
- P una distribución de probabilidad sobre los vértices en V generado por la estructura causal representada por G .
- $De(y_k) = \{v_j \in V \mid (v_j, y_k) \in E_j, E_j \in E, \forall j \ m \leq j \leq 1\}$ es un conjunto de todos los descendientes de y_k .
- $A \setminus \{b, c, d\}$, notación que expresa a todos los elementos del conjunto A que no incluye a los elementos b, c, d .

Condición de Markov Causal

G y P cumplen la condición si y solamente si para cada w en V , w es independiente de $V \setminus \{De(w) \cup Pa(w)\}$ dado sus padres $Pa(w)$.

Condición de Minimalidad

G y P cumplen la condición si y solamente si para cada subgrafo propio H de G con el conjunto de vértices V , el par $\langle H, P \rangle$ no satisface la condición de Markov. Un subgrafo propio tiene el mismo número de vértices, pero menos aristas entre vértices.

Condición de Confianza

Una distribución de probabilidad P sobre un grafo causal G que satisface la condición de Markov puede incluir otras relaciones de independencia, además de aquellas entalladas por la condición de Markov aplicada al grafo.

Si lo anterior no ocurre, las relaciones de independencia P son detalladas por la condición de Markov aplicada a G , se dice que P y G son confiables uno al otro [Glymour99, Spirtes01].

G y P cumplen la condición si y solamente si, cada relación de independencia condicional verdadera en P es inducida por la condición de Markov causal aplicada a G .

Separación-D

Pearl propuso una caracterización puramente gráfica, llamada separación-D, de independencia condicional, y probó que separación-D en sí, caracteriza todas las relaciones de independencia condicional que siguen de satisfacer la condición de Markov para un grafo dirigido acíclico [Pearl99, Glymour99, Spirtes01].

En la Tabla 2 se muestra los elementos para el método de Separación-D.

Tabla 2 Elementos de Separación-D

| | |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x, y | Vértices distintos en un grafo G . |
| W | Conjunto de vértices que no contienen a x, y . |
| Colinder | Un vértice v es un colinder sobre una ruta no dirigida U si y solo si hay dos aristas distintas sobre U que contienen a v como un vértice terminal y ambas inciden en v . |
| Ruta relativa activa a W | Cada vértice v en U es activo a W en caso de que: v es colinder y v o cualquiera de sus descendientes están en W o v no es colinder y no está en W . |

2.3.3 Algoritmo PC

El algoritmo PC tiene como objetivo encontrar el grafo causal para muestras lo suficientemente grandes. Dicho grafo representa las mismas condiciones de independencia condicional de la población bajo las siguientes suposiciones:

- El grafo causal en la población es acíclico
- El grafo causal en la población es confiable
- El conjunto de variables causales es causalmente suficiente

En el Algoritmo 3 se muestra el procedimiento general del algoritmo PC

Algoritmo 3 Proceso de creación de un Grafo Causal a través del Algoritmo PC.

Entrada: T = una tabla de datos con variables discretizadas correspondientes al conjunto V

Salida: Un grafo G que representa una estructura causal $C = (V, E)$

1. **Inicio**
2. $ADJ(x)$ = un conjunto de vértices adyacentes a x ; $x \in V$
3. $i=0$
4. **Formar** el grafo no dirigido completo G sobre V
5. **Repetir**
6. Para cada $x \in V$
7. Para cada $y \in ADJ(x)$
8. **Si** hay un subconjunto $W \subseteq ADJ(x) - \{y\}$ donde
9. $|W|=i$, x , y y son condicionalmente independientes dado el subconjunto W
10. **Entonces**
11. $W_{xy}=W$
12. **borrar** arista x - y
13. Fin para y
14. Fin para x
15. $i=i+1$
16. **Hasta** $|ADJ(x)| \leq i, \forall x$
17. **Para** cada tripleta de vértices x, y, z
18. **Si** $(x-y-z)$ y $(y \notin W_{xz})$ entonces
19. **orientar** x - y - z como $x \rightarrow y \leftarrow z$
20. $E_y = E_y \cup \{(y, x), (y, z)\}, Pa(y) = Pa(y) \cup \{x, y\}$
21. Fin para
22. **Repetir**
23. **Si** $x \rightarrow z -y$ Entonces
24. **orientar** z - y como $z \rightarrow y$
25. $E_y = E_y \cup \{(y, z)\}, Pa(y) = Pa(y) \cup \{z\}$
26. **Si** (hay una ruta dirigida de x a z) y (una arista entre ellos)
27. **Entonces**
28. **orienta** x - z como $x \rightarrow z$
29. $E_z = E_z \cup \{(z, x)\}, Pa(z) = Pa(z) \cup \{x\}$
30. **Hasta** que no haya más aristas por orientar
31. **Fin**

2.4 Índices de Caracterización

En esta sección se describirán algunas mediciones documentadas en la literatura, para caracterizar las instancias de entrada, muestras del espacio de soluciones, comportamiento y trayectoria del algoritmo, así también mediciones que permiten caracterizar el desempeño de algoritmo en su versión mono objetivo y multi objetivo.

En la entrada, los casos proporcionan información acerca de la estructura de las instancias del problema; en el proceso los elementos principales son aquellos que proporcionan ideas acerca del comportamiento del algoritmo en ejecución; y en la salida, se buscan índices importantes del desempeño. Después de identificar los elementos que intervienen en cada una de las fases del proceso de optimización, es necesario analizar qué aspectos es posible medir en cada categoría. Para esto es necesario estudiar tanto el problema como la implementación utilizada para resolverlo y seleccionar variables que revelen tendencias importantes.

2.4.1 Índices del Problema

En este apartado se dará introducción a los índices utilizados para la caracterización de las instancias. La información descriptiva de cada instancia del problema (*IDP*) se caracteriza por medio de índices que utilizan la información de los parámetros del problema de dicha instancia. A continuación se muestran índices que permiten la caracterización de la complejidad del problema. En la Tabla 3 se muestra el índice de la relación del tamaño del caso del problema a analizar.

Tabla 3 Índices del Tamaño del Problema.

| Nombre | Descripción | Formulación |
|-----------------------------------|------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Relación tamaño del caso [Cruz04] | Muestra la relación entre el tamaño del caso del problema y el caso más grande que ha sido resuelto. | $p = n/nmax$ <p>Donde:</p> <ul style="list-style-type: none"> • n: número de objetos • $nmax$: mejor número de elementos reportado en la literatura |

En la Tabla 4 se muestran índices propuestos en el trabajo de [Cruz04], que permiten la cuantificación de la relación entre los pesos de los objetos y el tamaño del contenedor.

Tabla 4 Índices de Relación entre los Pesos de los Objetos y el Tamaño del Contenedor.

| Nombre | Descripción | Formulación |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Proporción del tamaño/ contenedor [Cruz04] | Expresa la proporción del tamaño del caso total que se puede asignar en un contenedor de capacidad | $b = \begin{cases} 1 & \text{si } c > \sum_{i=1}^n s_i \\ \frac{c}{\sum_{i=1}^n s_i} & \text{en caso contrario} \end{cases}$ <p>Donde:</p> <p>s_i: Peso del objeto i c: Capacidad del contenedor n: número de objetos</p> |
| Capacidad ocupada del objeto [Cruz04] | Indica la capacidad ocupada por un objeto promedio en un contenedor | $t = \frac{\sum_{i=1}^n s_i / n}{c}$ <p>Donde:</p> <p>s_i: peso del objeto i n: número de objetos c: capacidad del contenedor</p> |
| Factor del peso [Cruz04] | Expresa la proporción de objetos cuyo tamaño s_i es factor de la capacidad del contenedor, donde: | $f = \frac{\sum_{i=1}^n \text{factor}(c, s_i)}{n}$ <p>Donde:</p> <p>s_i: peso del objeto i n: número de objetos c: capacidad del contenedor</p> |
| Dispersión de los pesos [Cruz04] | Indica el grado de la dispersión del cociente del tamaño de los objetos entre el tamaño de contenedor | $d = \sqrt{\frac{\sum_{i=1}^n \left[t - \frac{s_i}{c} \right]^2}{n}}$ <p>Donde:</p> <p>s_i: peso del objeto i n: número de objetos</p> |

| | | |
|--|--|--------------------------------------------------------------------------|
| | | c : capacidad del contenedor t : Capacidad ocupada del contenedor |
|--|--|--------------------------------------------------------------------------|

En la Tabla 5 se muestran índices que permiten la caracterización de la tendencia de los pesos de los objetos propuestos en el trabajo de [Alvarez06].

Tabla 5 Índices de Medidas de Tendencia Central de los Pesos.

| Nombre | Descripción | Formulación |
|-------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Media aritmética de los pesos [Alvarez06] | Calcula el valor esperado de un conjunto de pesos. | $ma = \frac{1}{n} \sum_{i=1}^n s_i$ Donde: s_i : peso del objeto i n : número de objetos |
| Media geométrica de los pesos [Alvarez06] | Calcula el promedio de los pesos, utilizando la multiplicación en vez de la suma. | $mg = \sqrt[n]{\prod_{i=1}^n s_i}$ Donde: s_i : peso del objeto i n : número de objetos |
| Media armónica de los pesos [Alvarez06] | Permite calcular el recíproco de la media aritmética del conjunto de pesos. | $mh = \frac{n}{\sum_{i=1}^n \frac{1}{s_i}}$ Donde: s_i : peso del objeto i n : número de objetos |
| Mediana de los pesos [Alvarez06] | Calcula el punto medio del conjunto de pesos. | $med = \frac{s_{(n+1)}}{2}$ Donde: s_{n+1} : posición del objeto $n+1$ n : número de objetos |
| Moda de los pesos [Alvarez06] | Es el valor que ocurre con más frecuencia en un conjunto de pesos. | $moda = \max(frecuencia(\sum_{i=1}^n s_i))$ Donde: s_i : peso del objeto i n : número de objetos |

En la Tabla 6 se muestran índices que permiten calcular la dispersión de los pesos pertenecientes a un conjunto de elementos, los cuales se documentan en el trabajo de [Alvarez06].

Tabla 6 Índices de Dispersión de Pesos: Rango, Error, Coeficiente de Variación, Varianza, Error Estándar.

| Nombre | Descripción | Formulación |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Rango [Alvarez06] | Es la diferencia entre el valor más grande y el valor más pequeño de un conjunto de datos. | $r = s_n - s_1$ Donde: s_n : peso del objeto n n : número de objetos |
| Desviación media de los pesos [Alvarez06] | Muestra la dispersión del peso de los objetos a través de la diferencia del valor absoluto entre cada peso de los objetos con respecto a la media del conjunto de pesos. | $dm = \frac{\sum_{i=1}^n s_i - \bar{s} }{n}$ Donde: s_i : peso del objeto i \bar{s} : es la media del peso de los objetos |

| | | |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | n : número de objetos |
| Varianza de los pesos [Alvarez06] | Mide la dispersión de peso de los objetos alrededor de la media. | $var = \frac{1}{n} \sum_{i=1}^n (s_i - \bar{s})^2$ <p>Donde: s_i: peso del objeto i \bar{s}: es la media del peso de los objetos n: número de objetos</p> |
| Desviación estándar de los pesos [Alvarez06] | Muestra que tan dispersos están los datos con respecto a la media. | $de = \sqrt{\frac{1}{n} \sum_{i=1}^n (s_i - \bar{s})^2}$ <p>Donde: s_i: peso del objeto i \bar{s}: es la media del peso de los objetos n: número de objetos</p> |
| Cuasivarianza [Alvarez06] | Calcula la dispersión o variabilidad de un conjunto de datos, permitiendo mostrar una estimación centrada de la varianza del conjunto de pesos. | $s2_nl = \frac{\sum_{i=1}^n s_i - \bar{s}^2}{n - 1}$ <p>Donde: s_i: peso del objeto i \bar{s}: es la media del peso de los objetos n: número de objetos</p> |
| Coefficiente de variación [Alvarez06] | Calcula la dispersión relativa del conjunto de datos. | $cv = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (s_i - \bar{s})^2}}{\left \frac{1}{n} \sum_{i=1}^n s_i \right }$ <p>Donde: s_i: peso del objeto i \bar{s}: es la media del peso de los objetos n: número de objetos</p> |
| Error estándar [Alvarez06] | Calcula la desviación estándar de una distribución muestral de un conjunto estadístico. | $e = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (s_i - \bar{s})^2}}{\sqrt{n}}$ <p>Donde: s_i: peso del objeto i \bar{s}: es la media del peso de los objetos n: número de objetos</p> |

En la Tabla 7 se muestran índices que permiten calcular la posición de los pesos propuestos en el trabajo de [Alvarez06].

Tabla 7 Índices de la Posición de los Pesos.

| Nombre | Descripción | Formulación |
|-----------------------|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cuartiles [Alvarez06] | Divide un conjunto de datos ordenados en cuatro partes porcentualmente iguales. | $c4 = L_i + \frac{\frac{\sum_{i=1}^n (N * k)}{4} - F_{i-1}}{f_i} * a_i$ <p>Donde: L_i es el límite inferior de la clase donde se encuentra el cuartil N: suma de las frecuencias absolutas F_{i-1} es la frecuencia acumulada anterior a la clase del cuartil a_i es la amplitud de la clase F_i: Frecuencia acumulada a la clase de cuartil</p> |
| Deciles [Alvarez06] | Divide un conjunto de datos ordenados en diez partes porcentualmente iguales. | $c10 = L_i + \frac{\frac{\sum_{i=1}^n (N * k)}{10} - F_{i-1}}{f_i} * a_i$ <p>L_i: límite inferior de la clase donde se encuentra el cuartil N: suma de las frecuencias absolutas F_{i-1} es la frecuencia acumulada anterior a la clase del cuartil</p> |

| | | |
|----------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | a_i es la amplitud de la clase F_i : Frecuencia acumulada a la clase de cuartil |
| Percentiles [Alvarez06] | Divide un conjunto de datos ordenados en cien partes porcentualmente iguales. | $c100 = L_i + \frac{\frac{\sum_{i=1}^n (N * k)}{100} - F_{i-1}}{f_i} * a_i$ L_i : limite inferior de la clase donde se encuentra el cuartil N : suma de las frecuencias absolutas F_{i-1} es la frecuencia acumulada anterior a la clase del cuartil a_i es la amplitud de la clase F_i : Frecuencia acumulada a la clase de cuartil |

En la Tabla 8 se muestran los índices que permiten el cálculo de la distribución de los pesos de los objetos en un contenedor

Tabla 8 Índices de la Forma de la Distribución de los Pesos.

| Nombre | Descripción | Formulación |
|-------------------------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Asimetría de Pearson basada en la mediana | Muestra la variación de Pearson basada en la mediana. | $amed = \frac{3(\bar{s} - mediana)}{de}$ Donde: \bar{s} : media del peso de los objetos $mediana$: la mediana del conjunto de pesos de : desviación estándar de s del conjunto de pesos de los objetos |
| Asimetría de Pearson basada en la moda | Muestra la variación de Pearson basada en la moda. | $amod = \frac{(\bar{s} - moda)}{de}$ Donde: \bar{s} es la media del peso de los objetos $moda$ representa la moda del conjunto de pesos de es la desviación estándar de s del conjunto de pesos de los objetos |
| Curtosis | Muestra la forma de la distribución de los tamaños de los objetos. | $curto = \frac{\sum_{i=1}^n (s_i - \bar{s})^4}{de^4}$ Donde: \bar{s} : media del peso de los objetos de : desviación estándar de s del conjunto de pesos de los objetos s_i : peso del objeto i |

Los índices anteriores (propuestos por Cruz y Álvarez) son retomados en los trabajos de Pérez y Landero [Pérez07, Landero08]. Los autores proponen, además, medir aspectos en cada etapa del proceso de optimización para analizar el desempeño de estrategias metaheurísticas aplicadas en la solución de BPP. Cada autor desarrolla métricas de desempeño para caracterizar el comportamiento de diferentes variantes del algoritmo de interés. En la Figura 1 se muestra los factores que se recomienda medir para cada fase

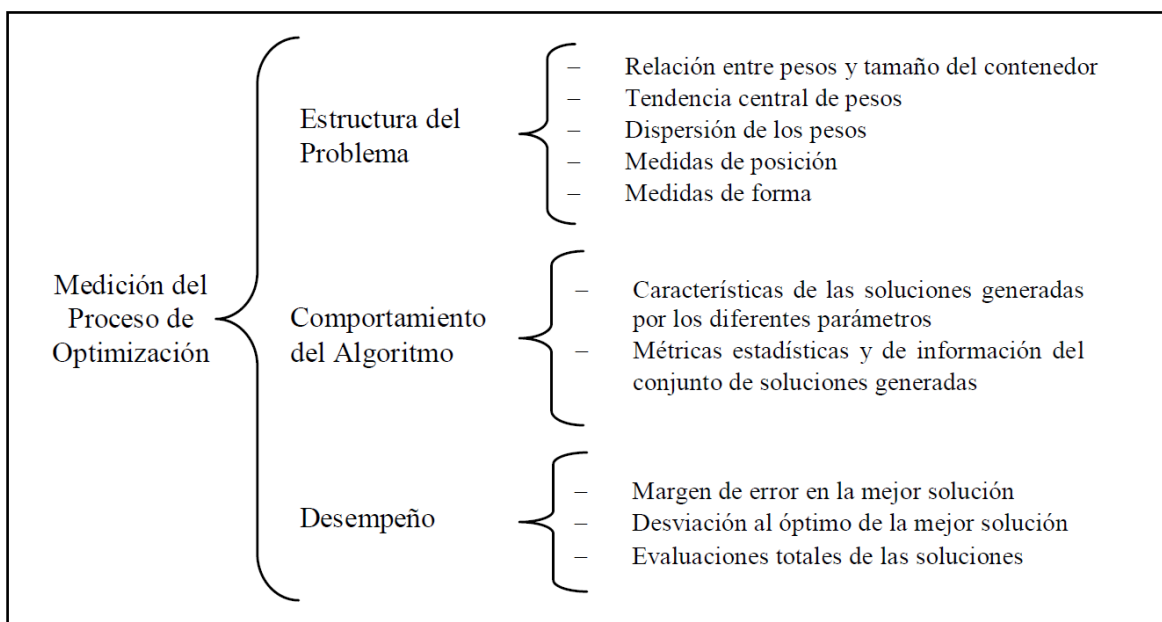


Figura 1 Clasificación del Proceso de Optimización.

2.4.2 Índices de la Muestra del Espacio de Soluciones

El espacio de soluciones de cada instancia del problema se muestrea antes de la experimentación con los algoritmos. En la Tabla 9 se muestra el índice de la muestra del espacio de soluciones el cual consiste en la generación de 100 soluciones aleatorias y cada una de estas soluciones es evaluada por la función aptitud $f(x)$.

Tabla 9 Índice de la Muestra del Espacio de Soluciones.

| Nombre | Descripción | Formulación |
|--------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Variación de los valores | Muestra la variación de los valores de la función aptitud de una muestra de soluciones. | $v_o = \frac{\sum_{i=1}^n (f(x_i) - \bar{f}x)^2}{m}$ <p>Donde: m: tamaño de la muestra $\bar{f}x$: es la media de los valores de la función aptitud f de las m soluciones</p> |

2.4.3 Índices del Comportamiento del Algoritmo

El comportamiento del algoritmo durante ejecución (*ICA*) es observado y caracterizado por los índices que se mencionan en la Tabla 10.

Tabla 10 Índices del Comportamiento del Algoritmo.

| Nombre | Descripción | Formulación |
|-------------------------|------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Media de las soluciones | Es el valor medio de un conjunto de las mejores soluciones de una población. | $m_b = \frac{1}{pob} \sum_{i=1}^{pob} f$ <p>Donde: pob: es el tamaño de la población f: es el desempeño de la mejor solución</p> |

| | | |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Varianza de las soluciones | Es la varianza del conjunto de las mejores soluciones aceptadas de una población. | $v_b = \frac{1}{n} \sum_{i=1}^n (f_i - \bar{f})^2$ |
| Porcentaje de soluciones aceptadas | Es el promedio del porcentaje de soluciones aceptadas por el algoritmo en un número de corridas | $pa = \frac{\sum_{i=1}^{ncor} pa_i}{ncor}$ |
| Porcentaje de soluciones rechazadas | Es el promedio del porcentaje de soluciones rechazadas por el algoritmo en un número de corridas | $pr = \frac{\sum_{i=1}^{ncor} pr_i}{ncor}$ |
| Coefficiente de autocorrelación | Índice que permite medir la rugosidad de una superficie de aptitudes, el valor 0 obtenido en el coeficiente de autocorrelación indica que los valores de la aptitud muestran poca correlación entre sí, y a un valor de 1 indica que la superficie es poco rugosa. | $p_k = \frac{\sum_{i=1}^{N-k} (f_i - \bar{f})(f_{i+k} - \bar{f})}{\sum_{i=1}^N (f_i - \bar{f})^2}$ |
| Longitud de autocorrelación | Índice que muestra el valor mayor de la distancia en la que el conjunto de soluciones se vuelve no correlacionado | $\ p_k\ = \frac{1}{1 - p_k}$ |

2.4.4 Índices de la Trayectoria del Algoritmo

La información de la trayectoria de búsqueda trazada por el algoritmo durante ejecución (*ITB*) ha sido caracterizada por varios índices, los cuales son descritos en [Merz99] y [Hoos04]. En el trabajo de [Lander08] menciona que la trayectoria de la búsqueda del algoritmo está construida por todos los valores de la función aptitud, generados por el algoritmo en orden de tiempo de ejecución. En la Tabla 11 se muestran algunos índices del proceso de caracterización de la trayectoria del algoritmo.

Tabla 11 Índices de la Trayectoria del Algoritmo.

| Nombre | Descripción | Formulación |
|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Número de puntos de inflexión en la trayectoria del algoritmo | Indica el cambio de dirección de los valores de la función aptitud de dos soluciones consecutivas de la trayectoria del algoritmo | $nc = \frac{\sum_{cor=1}^{ncor} nc_{cor}}{ncor}$ |
| Número de valles en la trayectoria del algoritmo | Índice el cual muestra la existencia de un valle en la trayectoria del algoritmo, cuando se encuentra una secuencia mayor de 5 soluciones donde su valor de la función aptitud va disminuyendo | $f(x_{i+1}) \leq f(x_i) 1 \leq i \leq (vx + i), y vx > 5$ $f(x_{vx+(j+1)}) \geq f(x_{vx+(j)}) 0 \leq j \leq vy, y vy \geq 1$ |
| Pendiente negativa del valle | Indica el promedio de las pendientes negativas de los valles encontrados en ncor corridas | $pn = \frac{\sum_{cor=1}^{ncor} pn_{cor}}{ncor}$ |
| Pendiente positiva del valle | Indica el promedio de las pendientes positivas de los valles encontrados en ncor corridas | $pp = \frac{\sum_{cor=1}^{ncor} pp_{cor}}{ncor}$ |
| Forma del valle | Índice el cual indica una aproximación de la forma del valle | $mp = \frac{pn + pp}{2}$ |

2.4.5 índices del Desempeño de Algoritmos Mono Objetivo

En esta sección se explican algunas mediciones propuestas en el trabajo de [Cruz04], [Alvarez06] y [Quiroz09], las cuales se muestran en la Tabla 12, para la caracterización del desempeño de algoritmos mono objetivo.

Tabla 12 Índices de la Caracterización del Desempeño de Algoritmos Mono Objetivo.

| Nombre | Descripción | Formulación |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Max_MejorF | Calcula el valor máximo de aptitud de una instancia. | $Max_MejorF = \max(M)$ |
| Min_MejorF | Este índice permite obtener el valor mínimo de aptitud de una instancia. | $Min_MejorF = \min(M)$ |
| Radio teórico | Este índice permite obtener la razón teórica donde Z_{enc} es el mejor valor obtenido en el algoritmo y Z_{opt} es el mejor valor reportado en la literatura para dicha instancia. | $Radio_teorico = \frac{Z_{enc}}{Z_{opt}}$ |
| Error obtenido | Este índice calcula el porcentaje de error del valor obtenido del algoritmo | $desv_apt = \frac{(Z_{opt} - Z_{enc})}{Z_{opt}}$ |
| Promedio de la función aptitud de las soluciones factibles | Índice muestra el promedio del desempeño de las soluciones factibles, en donde $Z_{factible}$ es el desempeño de una solución factible y pob es el tamaño de la población | $m_fact = \frac{\sum_{i=1}^{pob} Z_{factible}}{pob}$ |
| Varianza de la función aptitud de las soluciones factibles | Es la varianza de la aptitud de las soluciones factibles del algoritmo. | $var = \frac{1}{pob} \sum_{i=1}^{pob} (Z_{factible} - m_fact)^2$ |
| Coefficiente de engaño | Índice el cual permite evaluar la relación entre el problema y la trayectoria que describe un operador de búsqueda, si el valor obtenido muestra tendencia significativa a 1 significa que el problema es engañoso, mientras que un valor cercano al 0 indica que el problema es amigable o sencillo de resolver. | $coef_engaño = 1 - \frac{E - F_{min}}{F_{max} - F_{min}}$ |

2.4.6 Índices del Desempeño de Algoritmos Multi Objetivo

En la literatura se han documentado varias mediciones que permite la caracterización del desempeño de algoritmos que dan solución a problemas multi objetivo, el trabajo de Martínez [Martinez17] propone la evaluación del rendimiento de algoritmos que dan solución a problemas multi objetivo.

El trabajo de [Martinez17] clasifica las mediciones de desempeño en 3 categorías: la minimización de la distancia del frente de Pareto producido por el algoritmo con respecto al frente verdadero (suponiendo que se conoce), maximizar la distribución de soluciones obtenidas de tal manera que sea posible tener una distribución de vectores tan uniforme como sea posible y maximizar la cantidad de elementos del conjunto de óptimos de Pareto generados [Zitzler00].

En la Tabla 13 se muestra un enlistado de mediciones para la evaluación del desempeño de algoritmos multi objetivos el cual se encuentran en la literatura.

Tabla 13 Índices de la Caracterización del Desempeño de Algoritmos Multi Objetivo.

| Nombre | Función | Formulación |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Distancia Generacional [Van98] | <p>Mide la distancia de las soluciones que están en el conjunto de soluciones no dominadas encontradas hasta ahora de las del conjunto óptimo de Pareto</p> <p>Donde n es el número de soluciones en el conjunto de soluciones no dominadas encontrado hasta el momento y d_i es la distancia euclidiana (medido en el espacio objetivo) entre cada una de estas soluciones y el miembro más cercano del conjunto de Pareto óptima.</p> <p>Está claro que un valor de $GD = 0$ indica que todos los elementos generados están en el conjunto óptimo de Pareto. Con el fin de obtener resultados fiables, los conjuntos de soluciones no dominadas se normalizan antes de calcular la medida de distancia.</p> | $GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}$ |
| Distancia Generacional Invertida [Jiang14] | <p>Determina el promedio de las distancias entre cada solución del frente de Pareto verdadero a la solución más cercana del frente de Pareto encontrado. Este índice mide la proximidad que existe entre el frente de Pareto verdadero (FPV) y el frente de Pareto encontrado.</p> <p>Donde n es el número de soluciones en el Frente de Pareto verdadero, p es 2, y d_i es la distancia Euclidiana en el espacio de los objetivos entre cada vector de frente de Pareto verdadero y el punto más cercano del frente de Pareto encontrado.</p> <p>Si el resultado es 0 indica que ambos frentes son iguales; cualquier otro valor indica una divergencia entre ambos frentes; por ello es deseable obtener valores bajos de IGD.</p> | $IGD \triangleq \frac{(\sum_{i=1}^n d_i^p)^{\frac{1}{p}}}{n}$ |
| Distancia Euclidiana [Wang05] | <p>Esta métrica permite conocer la distancia que existe entre 2 soluciones obtenidas.</p> | $DE = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$ |
| Spread [Deb02] | <p>Es una métrica de diversidad que mide el grado de propagación logrado entre las soluciones obtenidas.</p> | $\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} d_i - \bar{d} }{d_f + d_l + (N-1)\bar{d}}$ |

| | | |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| | <p>Donde d_i es la Distancia Euclidiana entre N soluciones consecutivas, \bar{d} es la media de estas distancias, y d_f y d_l son las distancias euclidianas a las soluciones extremas (delimitador) del frente de Pareto exacta en el espacio objetivo.</p> <p>Esta métrica toma un valor de cero para una distribución ideal, señalando una extensión perfecta de las soluciones en el frente de Pareto. Se aplica esta métrica después de una normalización de los valores de la función objetivo.</p> | |
| Spacing [Liu07] | <p>Es una métrica el cual indica que tan bien las soluciones son distribuidas en el frente descubierto en donde n_{PF} es el número de miembros en PF_{know} y d'_i es la distancia euclidiana (en el dominio de los objetivos) entre el miembro i en PF_{know} y su miembro cercano en PF_{know}</p> | $s = \frac{\left[\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} (d'_i - \bar{d}')^2 \right]^{1/2}}{\bar{d}'}$ |
| Tasa de errores [Jiang14] | <p>Es una métrica la cual cuenta el número de soluciones óptimas en el frente de Pareto encontrado las cuales pertenecen al frente de Pareto verdadero y lo divide por el número de soluciones encontradas.</p> <p>El valor más bajo de esta medida muestra la mejor aproximación de las verdaderas soluciones óptimas de Pareto.</p> | $ER = \frac{\sum_{i=1}^n e_i}{n}$ |
| Hipervolumen [Zitzler99] | <p>Es un índice combinado de convergencia y diversidad que calcula el volumen (en el espacio objetivo) cubierto por los miembros de un conjunto no dominado de soluciones Q.</p> <p>Matemáticamente, para cada solución $i \in Q$, un hipercubo v_i se construye con un punto de referencia W y la solución i como las esquinas diagonales del hipercubo.</p> | $HV = volume \left(\bigcup_{i=1}^{ Q } (v_i) \right)$ |
| Cobertura de conjuntos [Jiang14] | <p>Presenta resultados del porcentaje de soluciones de un conjunto que son dominados por alguna solución de otro conjunto.</p> <p>$C(A, B)$ Proporción de soluciones en B, débilmente dominadas por soluciones en A. $C(A, B) = 1$ Todos los miembros de B son dominados débilmente por A. $C(A, B) = 0$ Ningún miembro de B es débilmente dominados por A.</p> | $C(A, B) = \frac{ \{b \in B \exists a \in A : a \preceq b\} }{ B }$ |
| Proporción de soluciones no dominadas [Tan02] | <p>Mide la proporción de las soluciones no dominadas de una población P. Para esto se obtiene el frente de Pareto (POF) de dicha población y se divide la cardinalidad de POF entre la cardinalidad de P.</p> | $RNI = \frac{ POF }{ P }$ |
| Proporción de hipervolumen [Van99] | <p>En caso de conocer el frente de Pareto óptimo teórico (POF*) se puede medir la proporción entre este y un POF obtenido esto se hace obteniendo el hipervolumen de ambos en base a un mismo punto de referencia.</p> | $HV - Ratio \triangleq \frac{HV(POF)}{hv(POF*)}$ |

2.5 Herramienta de Visualización de Análisis de Algoritmos Heurísticos (VisTHAA)

En este apartado se dará introducción de la herramienta a usar, se explicará cómo está estructurado internamente y además se explicará la funcionalidad de los módulos desarrollados actualmente.

Es una herramienta de diagnóstico dedicado al análisis de algoritmos heurísticos, esta herramienta permite a los investigadores extender sus funcionalidades a través de módulos integrados en una arquitectura para la facilitación de los análisis heurísticos. La herramienta cuenta con los siguientes módulos disponibles [Castillo11]:

1. Entrada de datos y preprocesamiento
2. Caracterización de instancias
3. Visualización de las instancias y el comportamiento del algoritmo
4. Visualización del espacio de búsqueda en tres dimensiones y el análisis de los algoritmos

En la Figura 2 se muestra la arquitectura interna de la herramienta VisTHAA, y los módulos que tienen una dependencia con otros módulos.

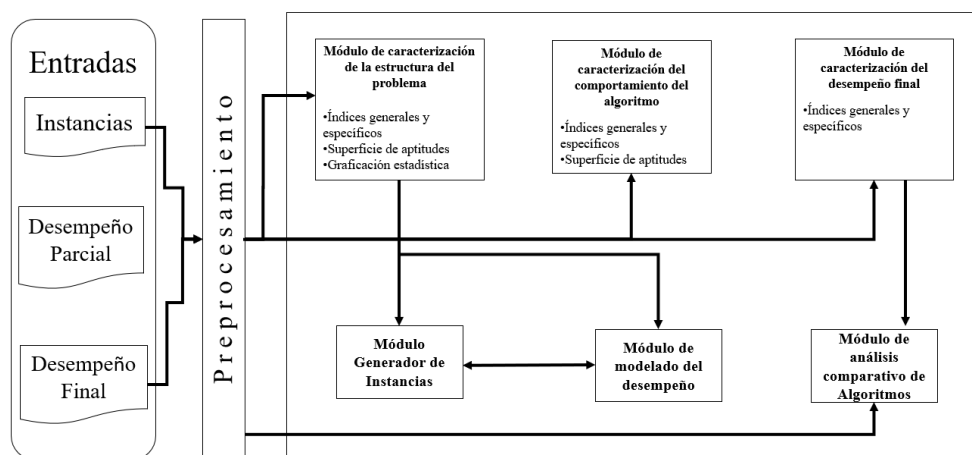


Figura 2 Arquitectura Interna de VisTHAA.

2.5.1 Entrada de Datos y Preprocesamiento

El módulo de entrada y preprocesamiento tiene como función que los investigadores utilicen sus propios formatos para la lectura y procesamiento de archivos, para cumplir esta función es necesario ajustar los archivos de entrada a un formato de entrada requerido por la herramienta.

El formato de instancias es descrito en un archivo (sin formato), esta descripción es conocida como metainstance. Posteriormente, el número de instancias a ser introducidos es específico, el nombre del archivo que describe las instancias y el nombre de cada una de ellas, esta especificación recibe el nombre de logbook. En la Figura 3 se muestra la ruta para el precargado del conjunto de instancias a trabajar y en la Figura 4 se muestra los archivos principales para el proceso de carga de instancias.

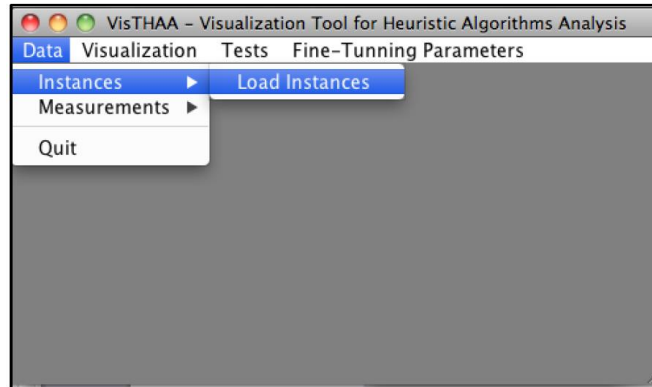


Figura 3 Ruta de Acceso para la Carga de Instancias [Castillo11].

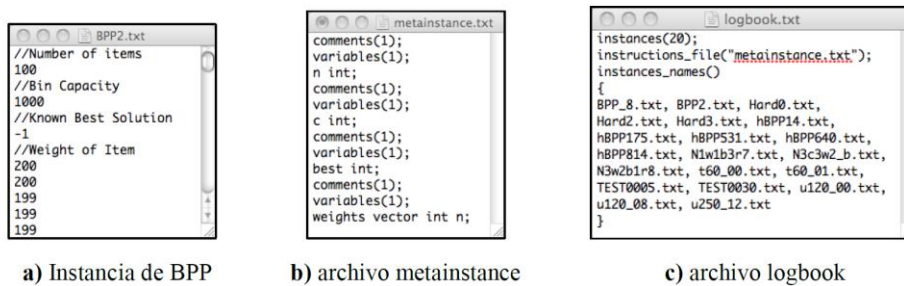


Figura 4 Visualización de los Elementos de la Entrada de Datos [Castillo11].

2.5.2 Caracterización de Instancias

Una vez finalizado el proceso de carga de información, el investigador puede aplicar un proceso de caracterización al conjunto de instancias del problema a analizar. Dicho proceso puede realizarse de manera estadística o visual. La caracterización estadística se realiza a través de la aplicación de índices al conjunto de instancias, el cual tiene como objetivo caracterizar y cuantificar factores que definen la estructura del problema, desempeño parcial y final del algoritmo, permitiendo la identificación de los factores que impactan en el proceso de optimización. La caracterización visual permite mostrar de manera gráfica al investigador, los resultados obtenidos de los índices aplicados en el proceso de caracterización, permitiendo una mejor comprensión de los mismos, y una comparación entre los resultados obtenidos y los que fueron mejores.

2.5.2.1 Módulo de Introducción de Nuevos Índices

Actualmente VisTHAA tiene implementados algunos de los índices especializados para BPP, los cuales pueden ser utilizados para obtener conocimiento de las instancias. Sin embargo, la herramienta ofrece a los investigadores la oportunidad de introducir sus propios índices a partir de las variables definidas en el archivo metainstance.

A manera de ejemplo, el índice que mide la capacidad del contenedor ocupada por un objeto promedio es introducido, el cual a su vez será utilizado en este estudio, en la Ecuación 1 se define el índice y en la Figura 5 se muestra una imagen de cómo debe de ser introducido a la herramienta.

$$t = \frac{\sum_{i=1}^n w_i / n}{c} \quad \text{Ecuación 1}$$

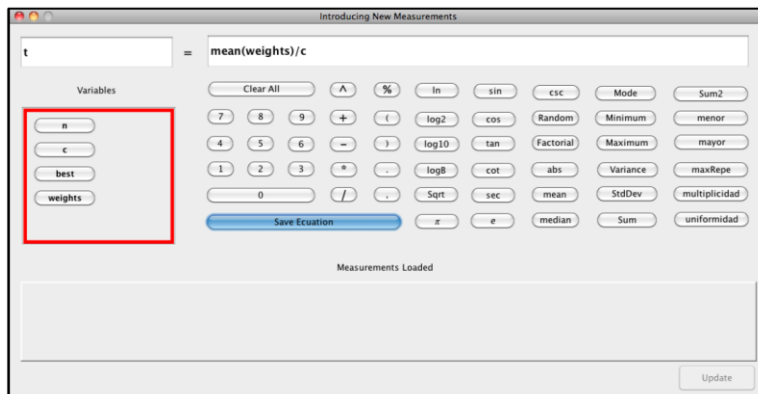


Figura 5 Módulo de Introducción de Nuevos Índices [Castillo11].

2.5.2.2 Módulo de Generación de Matriz de Características

Una vez que un conjunto de índices es introducido, el investigador puede decidir cuáles de ellos considerar para generar la matriz de características sobre el conjunto de instancias seleccionadas, el propósito es obtener información para el análisis de los valores calculados sobre dicho conjunto de instancias. En la Figura 6 se muestra el módulo que permite generar la matriz de características.

La matriz de características es una aportación importante en el trabajo de Castillo, ya que permite generar una matriz de datos, los cuales posteriormente pueden ser tratados con técnicas de análisis multivariado [Castillo11]. La matriz de características está conformada en sus filas por las unidades experimentales (instancias) y en sus columnas por las variables / atributos / características (índices que el investigador ingrese a VisTHAA).

La ventana para realizar los ajustes tiene en su lado izquierdo, un cuadro rojo, dentro del cual aparecen las instancias previamente cargadas a la herramienta, de las cuales el investigador puede seleccionar aquellas que desee incluir en la matriz.

En la parte central de la ventana, aparece un cuadro en color azul, dentro del cual el investigador puede seleccionar los índices que desee incluir en la matriz.

Finalmente, en el lado izquierdo de la ventana, aparecen dos opciones, las cuales tienen que ver con el archivo de salida de la matriz de características. Si se selecciona “Headers” el archivo de salida incluirá el nombre de los índices, si se selecciona “Instances names”, el archivo de salida incluirá los nombres de las unidades experimentales (instancias). Además, se encuentra un botón denominado “Calculate”, el cual genera la matriz de características y el archivo de salida en formato de texto plano, así como también muestra los resultados en otra ventana.

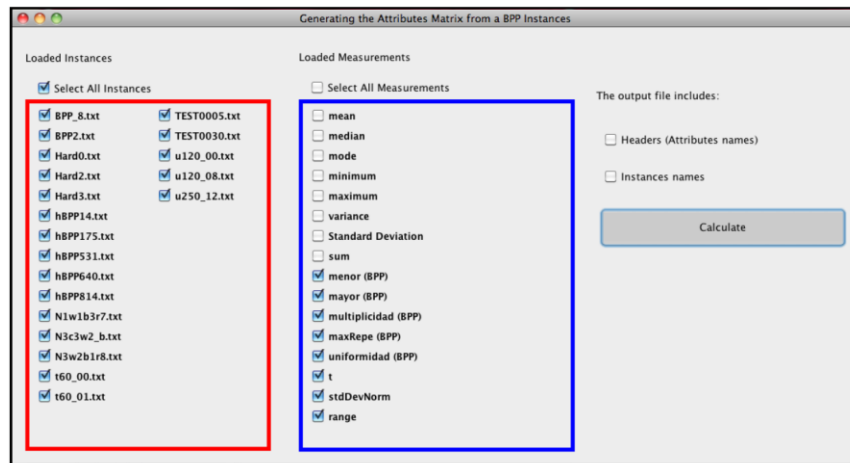


Figura 6 Módulo de Generación de Matriz de Características [Castillo11].

2.5.2.3 Módulo de Calculadora

Otra contribución que está estrechamente vinculada al módulo de caracterización es el uso de una calculadora, la cual muestra variables iniciales y mediciones de propósito general. Dicho módulo permite al investigador introducir una ecuación y obtener su resultado, sin que esta ecuación se conserve en la herramienta. En la Figura 7 se muestra un ejemplo de forma de calcular un índice nuevo.

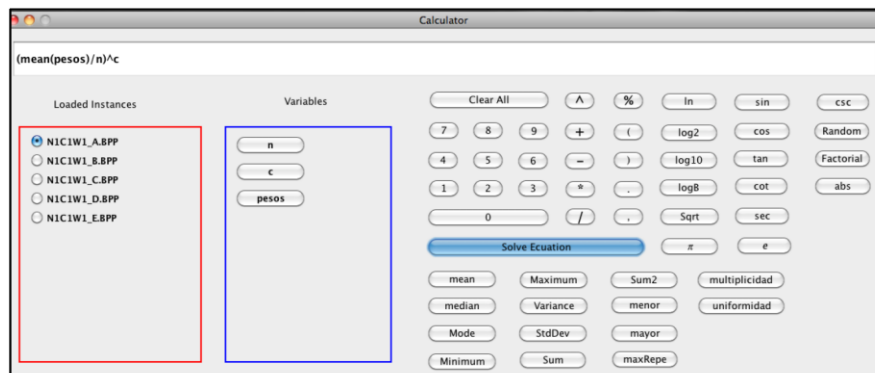


Figura 7 Ejemplo de Cálculo de un Índice Nuevo [Castillo11].

De la información almacenada el investigador se le puede permitir la introducción de nuevas mediciones en notación infija, además dicho módulo utiliza el algoritmo de Shunting Yard [Dijkstra61].

2.5.3 Visualización

El módulo de visualización tiene como objetivo el poder visualizar de manera gráfica las instancias cargadas en la herramienta, graficas de frecuencia de las instancias, graficas del paisaje de aptitud del comportamiento del algoritmo y graficas estadísticas.

2.5.3.1 Instancias del Problema

La visualización de la información de las instancias del problema a estudiar se realiza a través de una gráfica de barras bidimensional. En dicha estrategia, la visualización de las instancias se realiza a través de barras, donde el eje de las abscisas representa al i -ésimo objeto, mientras que el eje de las

ordenadas representa el peso de los objetos. En la Figura 8 se muestra la ruta del proceso de visualización en VisTHAA.

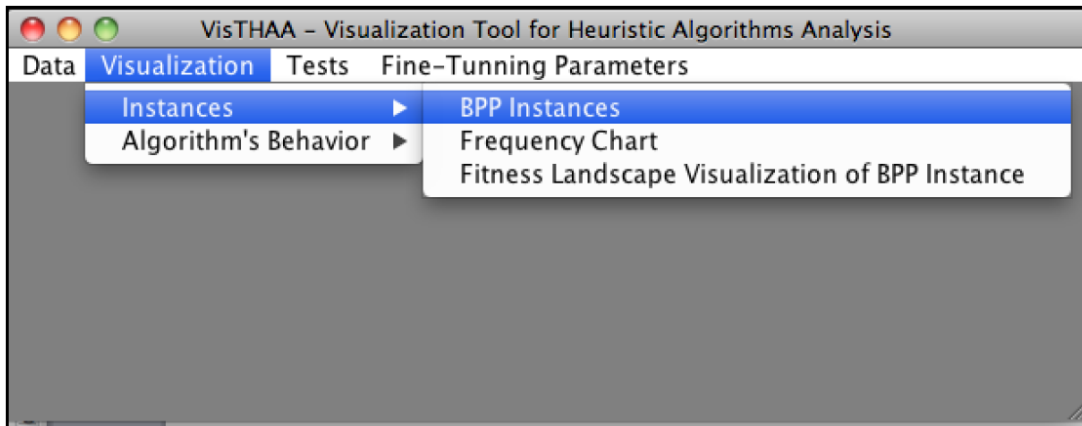


Figura 8 Ruta del Módulo de Visualización de Instancias [Castillo11].

Al seleccionar el módulo de visualización de instancias, aparecerá una ventana en la cual se deberá seleccionar una de las instancias cargadas, con el fin de visualizar la información de la instancia o también mostrar la información de la instancia en forma ascendente, en la Figura 9 se muestra el módulo de visualización de instancias.

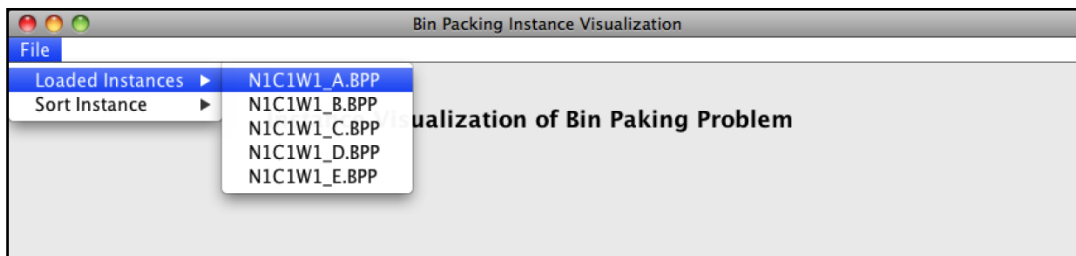


Figura 9 Módulo de Visualización de Instancias de VisTHAA [Castillo11].

Una vez seleccionada la instancia a visualizar, la herramienta procede a visualizar la información de la instancia en una gráfica de barras como se muestra en la Figura 10.

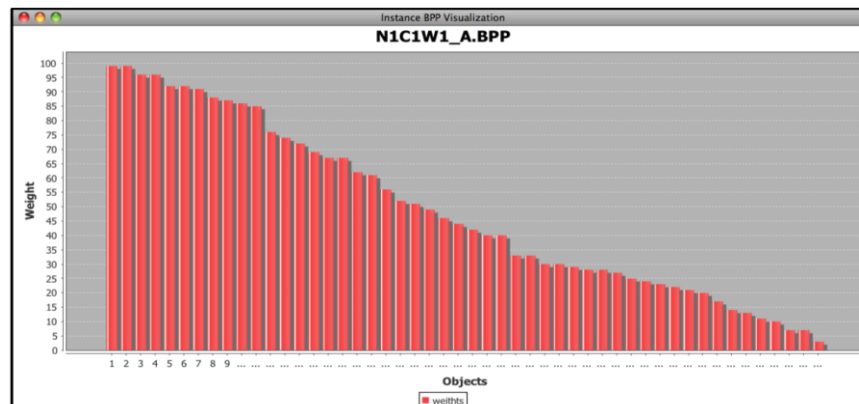


Figura 10 Visualización Gráfica de los datos de Instancias Cargadas [Castillo11].

2.5.3.2 Instancias del Problema de Orden Ascendente

En la segunda opción mencionada en el punto anterior, la herramienta permite visualizar la información de las instancias cargadas, en la Figura 11 se muestra la selección de las instancias que se desea visualizar de forma ascendente.

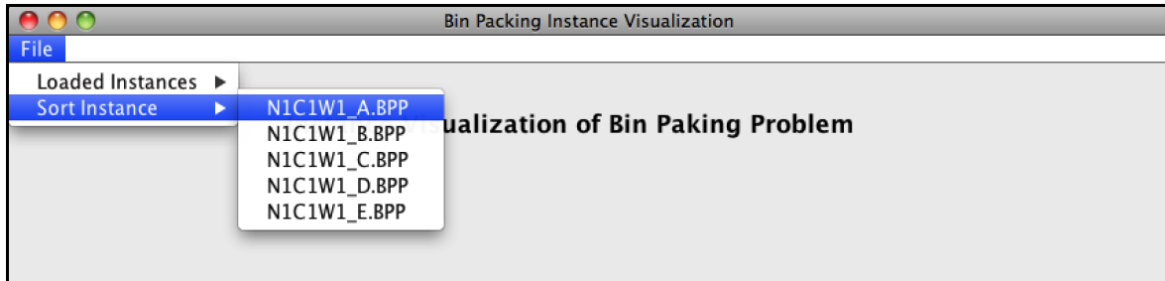


Figura 11 Módulo de Visualización de Instancias en forma Ascendente de VisTHAA [Castillo11].

Una vez seleccionada la instancia a visualizar, la herramienta procede a visualizar la información de la instancia en una gráfica de barras en forma ascendente como se muestra en la Figura 12.

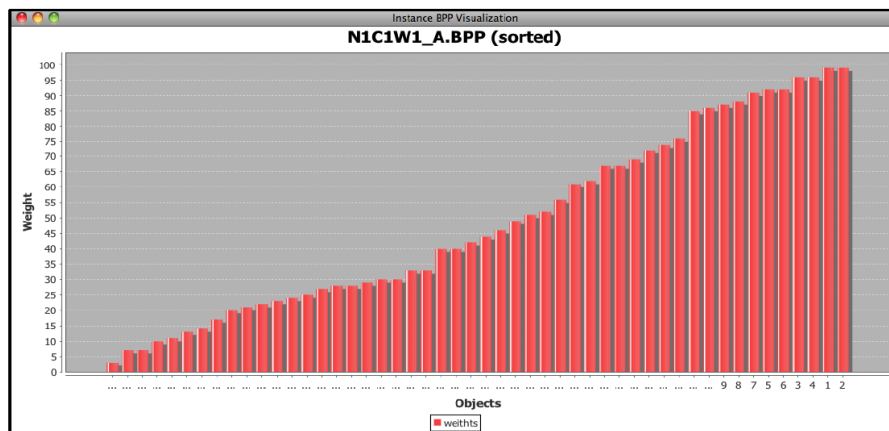


Figura 12 Visualización Gráfica de los Datos de Instancias Cargadas en forma Ascendente [Castillo11].

2.5.3.3 Graficas de Frecuencia

Este tipo de gráficas permiten conocer la cantidad de objetos que se repiten dentro de un determinado porcentaje de la capacidad del contenedor y dentro de que rango se encuentran los objetos. En la Figura 13 se muestra la ruta del proceso de visualización de graficas de frecuencias de VisTHAA. El eje de las abscisas representa el peso de los objetos como porcentaje de la capacidad del contenedor, es decir $0 < w_i/c \leq 1$.

El eje vertical cuantifica el número de objetos que se localizan dentro de un intervalo de porcentaje [Quiroz 09]. El porcentaje de incremento es del 1%, lo que significa que se contabilizará cuantos objetos están entre el 0% y el 1%, cuantos objetos están entre el 1% y el 2% y así sucesivamente hasta llegar al 100%.

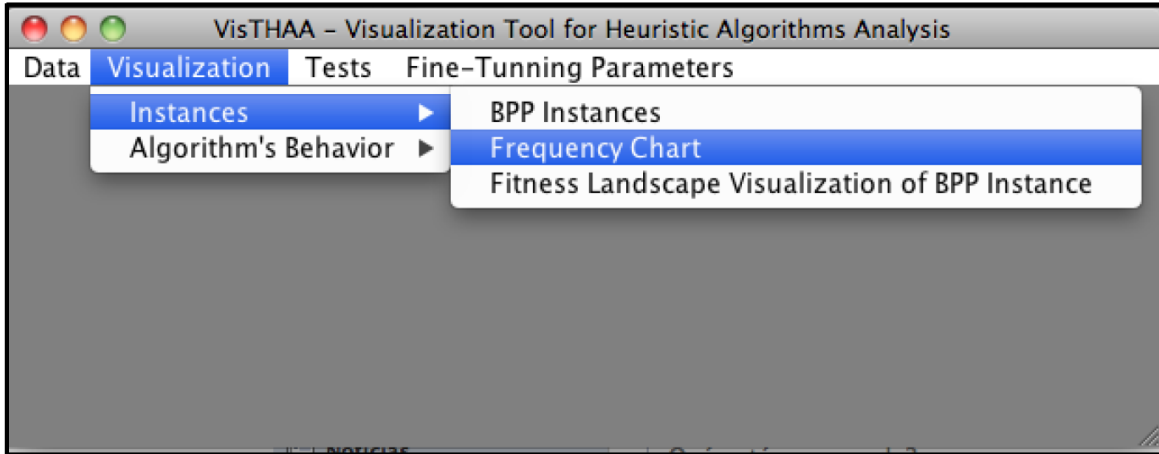


Figura 13 Ruta del Módulo de Visualización de Gráficas de Frecuencia [Castillo11].

Una vez accedido al módulo lo que continúa es seleccionar la instancia específica a ser visualizada, la cual se muestra en la Figura 14.

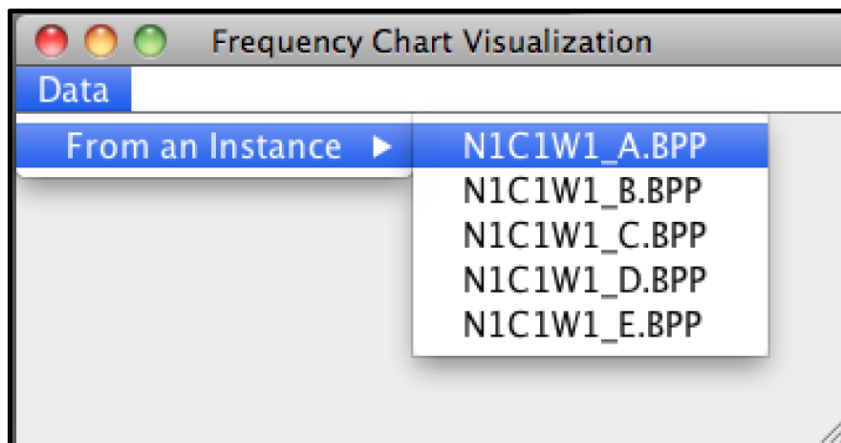


Figura 14 Módulo de Visualización de Gráficas de Frecuencia VisTHAA [Castillo11].

Posterior a la selección de la instancia, se muestra la gráfica de frecuencias referente a ella, la cual se muestra en la Figura 15.

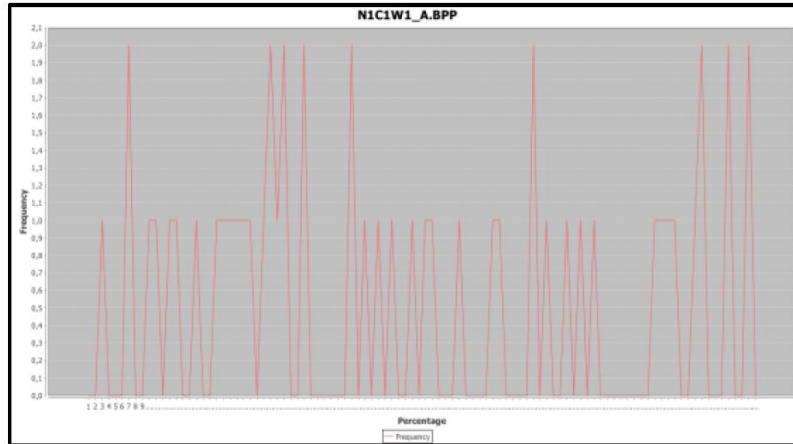


Figura 15 Visualización Gráfica del Peso de los Elementos de la Instancia N1C1W1_A.BPP [Castillo11].

2.5.3.4 Graficas de Superficie de Aptitudes

En la tercera opción mencionada en el punto anterior, la herramienta permite visualizar la superficie de aptitudes a través de una caminata aleatoria. En la Figura 16 se muestra la ruta del módulo de visualización de superficie de aptitudes.

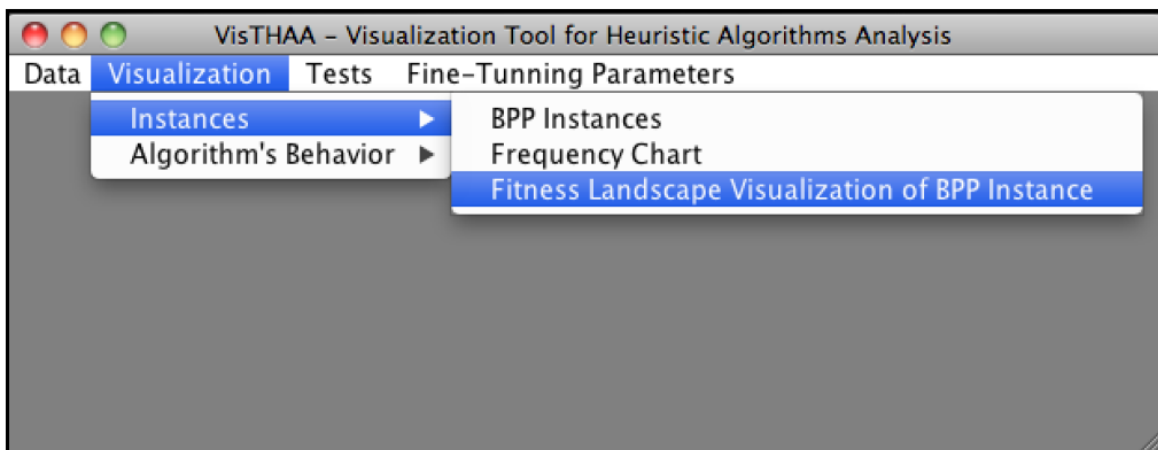


Figura 16 Ruta del Módulo de Visualización de la Superficie de Aptitudes [Castillo11].

La visualización de la superficie de aptitudes se llevó al siguiente nivel, visualizarla en las tres dimensiones espaciales. La idea general del método es extraer una muestra representativa de la instancia a través de un algoritmo Random Walk. Luego, ese conjunto de datos unidimensional convertirlo en un conjunto bidimensional sin alterar de ninguna manera los valores de aptitud calculados. Finalmente, ya con tres variables por cada solución (valor de fila, valor de columna y valor de aptitud) es posible visualizar dichos puntos en el espacio.

Descripción general del Algoritmo Random Walk

La idea general del algoritmo *Random Walk* es obtener una solución inicial factible, y a partir de ella generar soluciones vecinas iterativamente hasta un cierto número de pasos predefinidos.

En cada paso de la caminata aleatoria una nueva solución es evaluada, sin embargo, no es de interés que el algoritmo reporte la mejor solución alcanzada durante su ejecución, dicho algoritmo fue implementado en el trabajo de [Castillo11] el cual se muestra a continuación.

1. **Generar** una solución inicial factible.
2. **Computar** los valores de la función objetivo y la función de aptitud.
3. **Para** $i = 1$ hasta $N - 1$, hacer:

Generar una solución vecina de la solución actual.

Computar los valores de la función objetivo y la función de aptitud.

4. **Almacenar** los resultados en un archivo de texto plano.

Donde:

N es el número de pasos de la caminata aleatoria.

En la Figura 17 se observa en la parte izquierda de la ventana que aparecen las instancias que han sido cargadas a la herramienta, en la parte central algunos campos deben de ser configurados para la ejecución del algoritmo, una vez completado todos los campos se procede a hacer clic en el botón de “Execute Random Walk”.

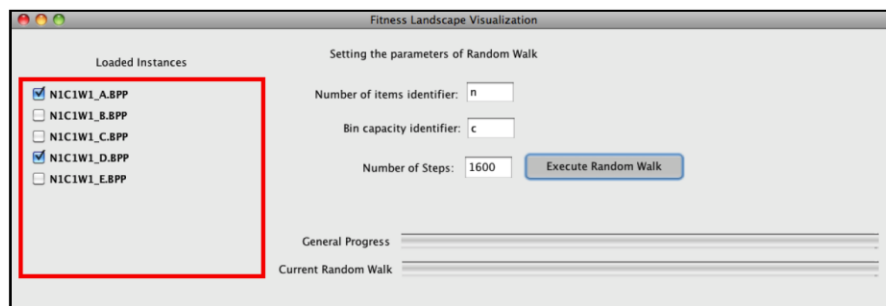


Figura 17 Módulo de Visualización de la Superficie de Aptitudes [Castillo11].

En la Figura 19 se muestra una gráfica en 3D de la caminata aleatoria aplicado a una instancia del problema de empaqueo de contenedores.

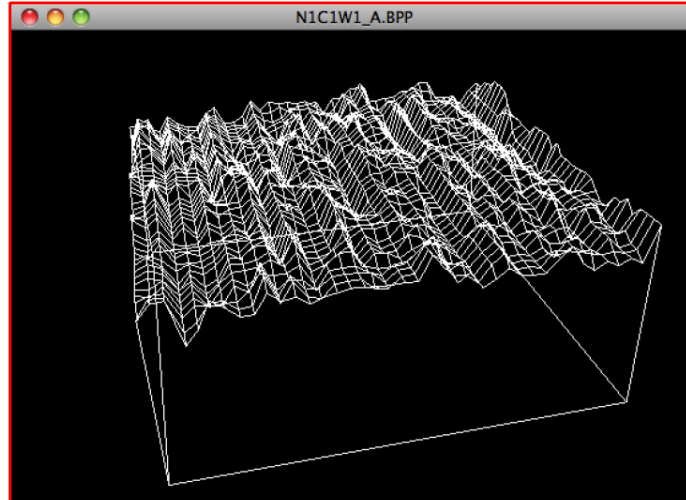


Figura 18 Visualización de la Superficie para la Instancia N1C1W1_A.BPP [Castillo11].

2.5.4 Módulo de Análisis Estadístico

La función de este módulo permite la evaluación de desempeño de dos algoritmos, dicha evaluación es efectuada a través de la prueba de Wilcoxon, el cual es clasificado como un procedimiento no paramétrico que determina si dos conjuntos de datos muestran una diferencia significativa [Mendenhall97].

La prueba de Wilcoxon se utiliza para determinar si dos muestras representan dos diferentes poblaciones [García 08]. Específicamente en el ámbito de las ciencias computacionales, la prueba de Wilcoxon se utiliza para determinar si las diferencias en el desempeño de dos algoritmos son estadísticamente significativas, ya que, de ser así, se puede afirmar fehacientemente, con un nivel de confianza determinado, que un algoritmo es superior a otro.

Debido a lo anterior, se implementó un módulo en la herramienta VisTHAA que permitiera determinar si un algoritmo es superior a otro estadísticamente, a través de la aplicación de la prueba de Wilcoxon. En la Figura 19 se muestra la ruta del módulo de prueba estadística de Wilcoxon.

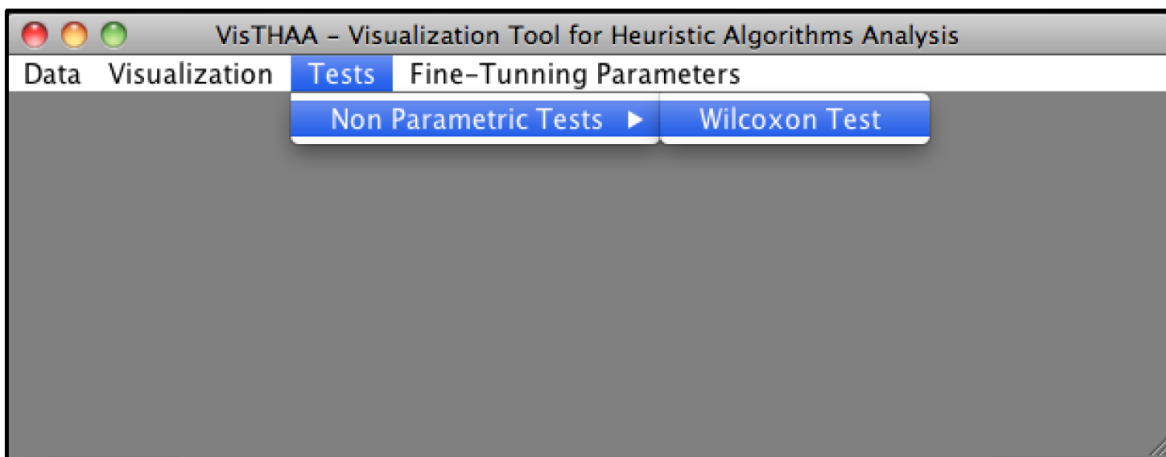


Figura 19 Ruta del Módulo de Prueba de Wilcoxon [Castillo11].

Una vez accedido al módulo, se muestra una ventana cuyo principal objetivo es cargar a VisTHAA los datos del desempeño de los algoritmos, para esto accedemos al menú *Load*, y luego al menú *...from a File*, tal y como se muestra en la Figura 20.

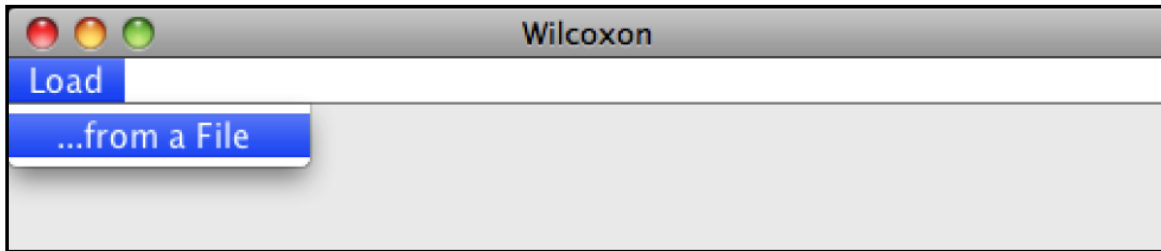


Figura 20 Módulo de Prueba Paramétrica de Wilcoxon [Castillo11].

En la Figura 21 se muestra un navegador de archivos, dentro del cual, al igual que las instancias, debemos buscar el archivo correspondiente, en este caso, el archivo donde se encuentren los datos de desempeño de los algoritmos a ser evaluados.

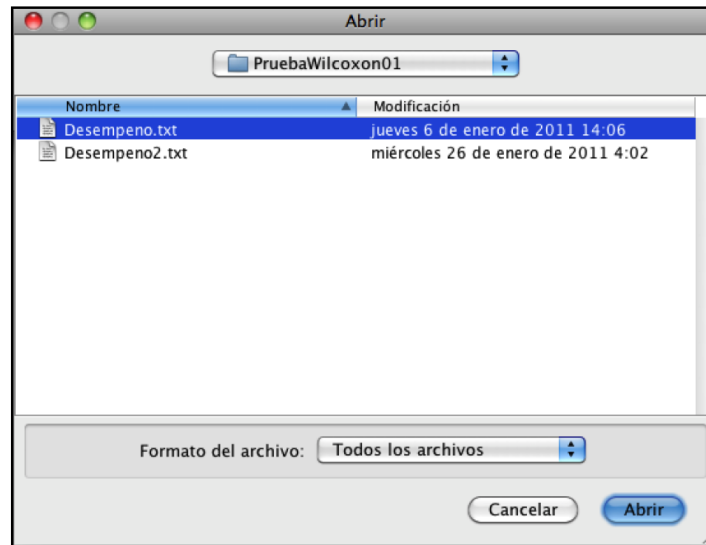


Figura 21 Ventana de Búsqueda de Archivo de Desempeño [Castillo11].

Cabe mencionar que los datos de desempeño se toman de un archivo de texto plano el cual únicamente debe tener dos líneas, donde la primera línea debe contener un conjunto de datos del desempeño del primer algoritmo, mientras que en la segunda línea debe contener los datos del desempeño del segundo algoritmo.

Una vez hecho esto, una nueva ventana es mostrada, ya con los datos del archivo seleccionado, ahora únicamente hace falta hacer clic en el botón *Execute Wilcoxon Test* para que VisTHAA automáticamente realice dicha prueba. La Figura 22 muestra la ventana principal de la prueba de Wilcoxon.

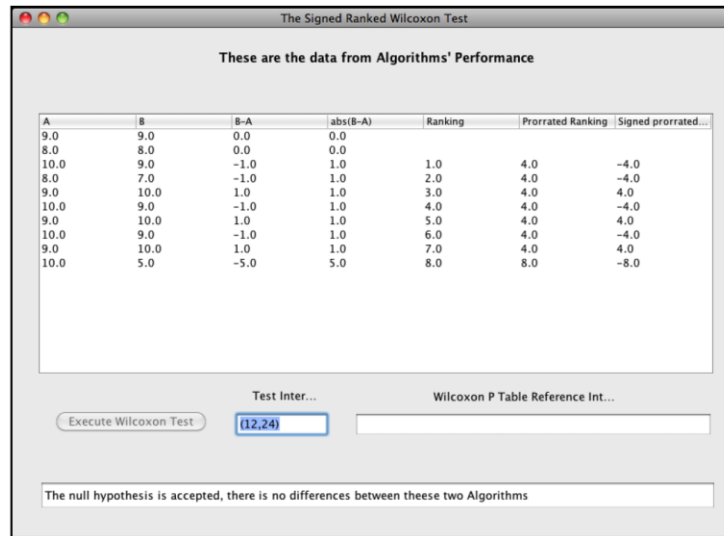


Figura 22 Ventana de Resultados de la Prueba de Wilcoxon [Castillo11].

Capítulo 3: Propuesta de Solución

En este capítulo se presenta la metodología propuesta para el cumplimiento de los objetivos de este trabajo de tesis, el cual se conforma por un subconjunto de procesos que permitirán el mejoramiento del desempeño de algoritmos mono/multi objetivo a través de técnicas causales.

Para la aplicación de dicha metodología, se necesita información obtenida de índices que permiten la caracterización de la estructura del problema, del espacio de soluciones, del comportamiento, trayectoria y desempeño del algoritmo solucionador para la elaboración del rediseño causal.

Previamente la herramienta VisTHAA no contaba con un módulo ni métodos para llevar a cabo el análisis causal, el cual necesita información de instancias, algoritmos e índices para mejorar la calidad del desempeño a través del rediseño causal.

Las aportaciones realizadas en este trabajo para la herramienta VisTHAA se describen a continuación:

1. Análisis y rediseño del módulo de preprocesamiento de instancias para problemas multi objetivo y ampliación del preprocesamiento de instancias mono objetivo, pertenecientes a la familia de contenedores.
2. Análisis y rediseño del módulo de caracterización de la estructura del problema, para la incorporación de instancias multi objetivo y la incorporación de más índices de la estructura problema.
3. Análisis, diseño e implementación del módulo de caracterización del comportamiento algorítmico e incorporación de índices del comportamiento algorítmico mono/multi objetivo.
4. Análisis, diseño e implementación del módulo de caracterización del desempeño de algoritmos e incorporación de índices del desempeño algorítmico mono/multi objetivo.
5. Análisis, diseño e implementación de la metodología causal propuesta por [Lander08] para algoritmos multi objetivo, e implementación del módulo de rediseño causal de algoritmos mono/multi objetivo en VisTHAA.

En la Figura 23 se muestra la nueva arquitectura de VisTHAA, resaltando con borde punteado los módulos propuestos en este trabajo de tesis. Dichos módulos se describen a lo largo de este capítulo.

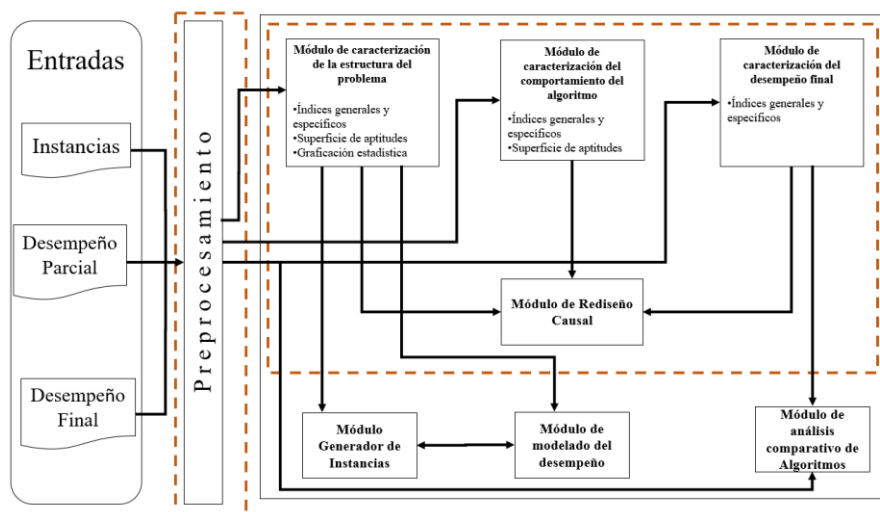


Figura 23 Diagrama General de la Arquitectura de VisTHAA.

3.1 Metodología del Módulo del Preprocesamiento

En esta sección se da explicación de la metodología propuesta para el rediseño del módulo de preprocesamiento de instancias para la incorporación de nuevos problemas mono objetivo y la incorporación de instancias multi objetivo, tomando como base el trabajo de [Castillo11]. En la Figura 24 se resalta con borde punteado la sección en donde se aplicó el rediseño del módulo de preprocesamiento para la aceptación de instancias multi objetivo y más instancias mono objetivo.

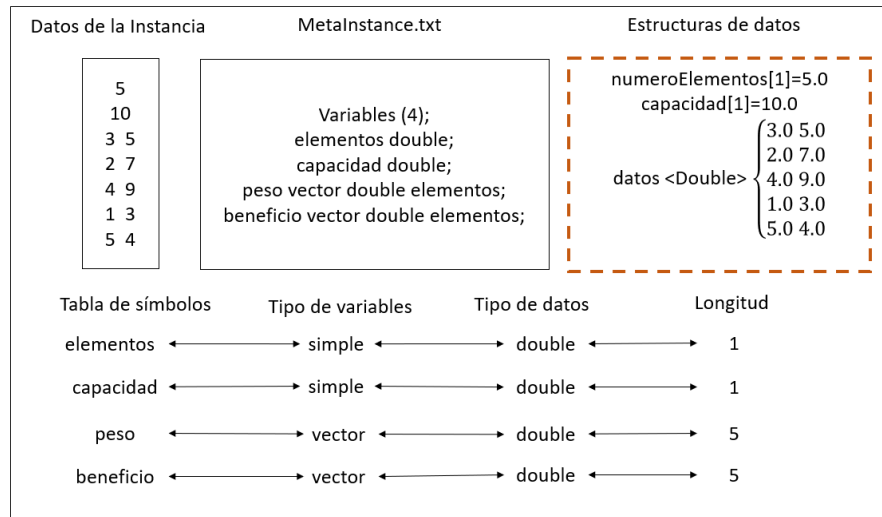


Figura 24 Ejemplo del Proceso de Almacenamiento de Instancias del Problema de la Mochila en VisTHAA.

Como se muestra en la Figura 24, los elementos que conforman el módulo de preprocesamiento de la herramienta VisTHAA, los cuales son el *conjunto de instancias* a procesar, el *archivo de MetaInstance* el cual contiene los atributos que harán referencia a la información del conjunto de instancias procesadas, *estructuras de datos* en donde se almacena la información de las instancias procesadas y la *tabla de símbolos* el cual almacena los atributos definidos en el archivo *MetaInstance*.

En este trabajo se propone el rediseño de la sección estructuras de datos para la incorporación de nuevas instancias mono / multi objetivo, el cual se encuentra resaltado en línea punteada en la Figura 24. En el Algoritmo 4 se muestra el seguimiento del proceso de carga de instancias actual de VisTHAA para problemas mono/multi objetivos.

Algoritmo 4 Pasos del Preprocesamiento de Instancias.

1. **Inicio**
2. **Llamar** el método *GetLogbook()*
3. **Guarda** el nombre del archivo logbook→str
4. **Llamar** el método *CompileLogbook(str)*
5. **Compila** el archivo logbook
6. **Llamar** el método *InstancesFunctions(str)*
7. **Guarda** el número de instancias a preprocesar
8. **Llamar** el método *InstructionsFileFunction(str)*
9. **Guarda** el nombre del archivo metainstance
10. **Llamar** el método *InstancesNamesFunction(str)*
11. **Guardar** la ruta y el nombre del conjunto de instancias a preprocesar →instances
12. **Llamar** el método *GetMetainstance()*
13. **Lee** las variables del archivo *metainstance*
14. **Llamar** el método *CalculateNumberOfVariables()*

15. **Guarda** el número de variables → *n_variables*
16. **Llamar** el método *ResizeStructures(n_variables)*
17. **Redimensiona** del conjunto de estructuras de almacenamiento del problema
18. **Llamar** el método *LoadInstance(instances)*
19. **Guarda** la información del conjunto de instancias a procesar en las estructuras de almacenamiento
20. **Llamar** el método *ShowInstances(n_variables)*
21. **Imprime** en pantalla las instancias cargadas
22. **Fin**

A continuación en la Tabla 14 se describen los métodos que conforman el módulo de preprocesamiento actual de la herramienta VisTHAA, incluyendo los que fueron desarrolladas en el trabajo de [Castillo11], y resaltando en negritas los que se rediseñaron en este trabajo de tesis.

Tabla 14 Listado de Métodos del Módulo de Preprocesamiento de VisTHAA.

| Nombre del método | Descripción |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLogbook() | Lee el archivo "logbook.txt" y hace un preprocesamiento en el cual consiste en la concatenación de caracteres dentro del archivo |
| InstructionsFileFunction() | Obtiene el nombre del archivo de instrucciones llamado "metainstance.txt" mediante una lectura en base de tokens, en el archivo "logbook.txt" |
| InstancesNamesFunction() | Obtiene el nombre de la instancia o instancias con las que se trabajara mediante la lectura en base de tokens |
| InstancesFunctions() | Obtiene la cantidad de instancias ignorando las cadenas y los paréntesis mediante una lectura en base de tokens, en el archivo "logbook.txt" |
| CompileLogbook() | Permite realizar la compilación de archivo "logbook.txt". este proceso consta de los siguientes pasos: 1.- Separar las instrucciones (separador = ";") 2.- Leer cada instrucción hasta el paréntesis apertura 3.- En base a la instrucción leída, verificar que sea una del diccionario 4.- Si es una palabra reconocida por el diccionario, entonces: 4.1.- Obtener el tipo de argumento a ser leído y almacenarla en la variable correspondiente 4.2.- si no, mostrar el mensaje de error correspondiente y salir |
| GetMetainstance() | Permite leer el archivo "metainstance.txt" mediante una lectura en base de tokens e identificando la cantidad de variables checando los símbolos, saltos de línea y palabras reservadas para comprobar que están en el diccionario. Una vez que el sistema encuentre un símbolo ";" comprueba si es la última instrucción y si no lo es, guarda la variable y el tipo de variables de la siguiente línea en el sistema para su uso posterior. |
| CalculateNumberOfVariables() | Método mediante el cual obtiene la cantidad de variables del archivo "metainstance.txt", que se especifica en la primera línea, checa si la instrucción contiene la palabra reservada "variables" entonces se obtiene el valor que se encuentra dentro de los tokens "(" y ")" y lo devuelve como un entero. |
| ResizeStructures() | Método en el cual se redimensionan las estructuras de almacenamiento en las nuevas dimensiones obtenidas para el problema de empaquetado, en base de los métodos: 1.- Func_instancias() 2.- CalculaCantVariables() |
| LoadInstance() | Método por el cual se cargan las instancias especificadas en el archivo "logbook.txt" a las estructuras de control dentro de la herramienta aplicado para el problema de empaquetado. |
| ShowInstances() | Método en el cual se muestra en pantalla las instancias que han sido cargadas. |
| Main() | Método principal en el cual se llaman una serie de funciones que tiene como función el preprocesamiento de la herramienta |

Los métodos resaltados en la Tabla 14 fueron propuestos a rediseñar, debido a que cumplen con la función del redimensionamiento de las estructuras de guardado (**ResizeStructures**) y el proceso de

almacenamiento de las instancias (LoadInstance) de los problemas a analizar propuestos en el capítulo 2.

3.2 Metodología del Módulo del Rediseño Causal

En esta sección se presenta la metodología propuesta para el mejoramiento del desempeño de algoritmos mono/ multi objetivo, a través del rediseño causal tomando como base la metodología propuesta en el trabajo de [Lander08], dicho trabajo ha sido probado para algoritmos mono/multi objetivo.

En la Figura 25 se muestra la metodología a seguir para el rediseño de algoritmos, el cual está conformado por una serie de fases y elementos, las cuales se explican a lo largo de este trabajo de tesis.

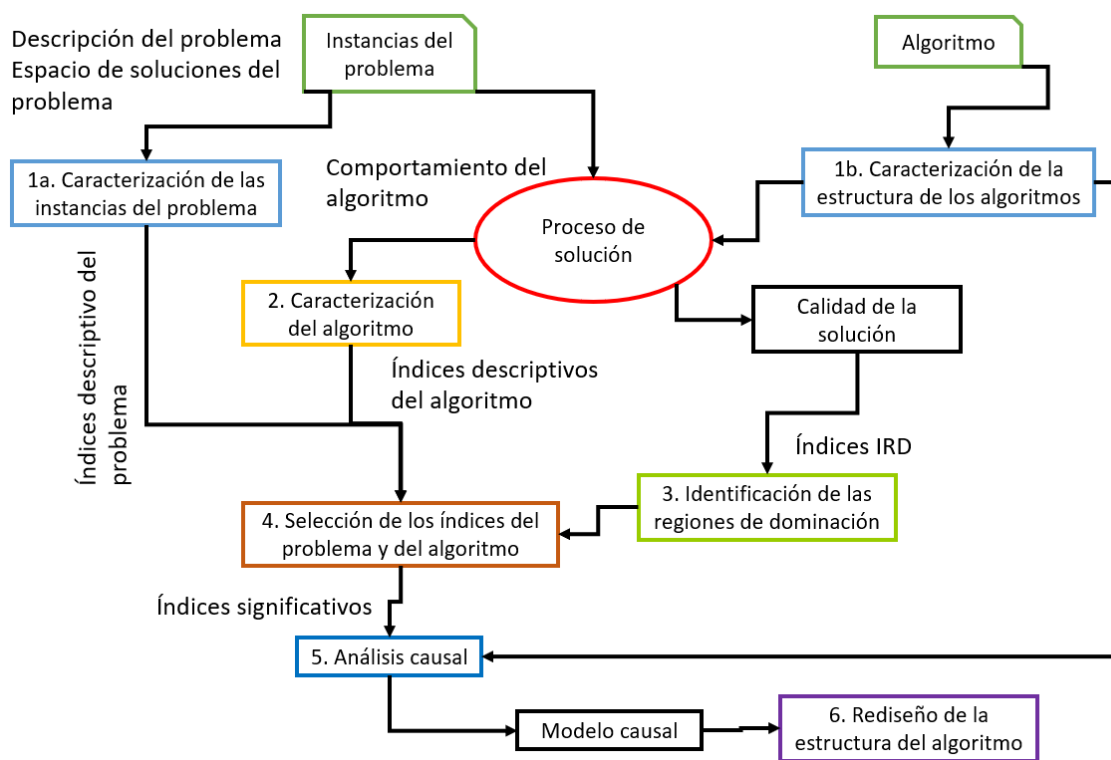


Figura 25 Metodología del Rediseño Causal de Algoritmos Mono/Multi Objetivo.

Los elementos necesarios para llevar a cabo la ejecución de la metodología mostrada en la Figura 26 son: el conjunto de instancias de entrada pertenecientes al problema a analizar y el conjunto de algoritmos a rediseñar. A continuación se da una descripción de las fases involucradas en la metodología del rediseño causal.

Instancias del Problema y el Algoritmo Solucionador

Esta fase tiene como objetivo la definición del conjunto de instancias de datos pertenecientes al problema a analizar y el algoritmo solucionador que proporcionará resultados del conjunto de instancias del problema para la toma de decisiones.

Proceso de Caracterización

En esta fase se tiene como objetivo el cuantificar los siguientes procesos: la estructura del problema, muestra del espacio de soluciones, comportamiento, trayectoria y desempeño del algoritmo a través de índices descritos en el capítulo 2. A continuación se da una descripción de los tres subprocesos del proceso de caracterización mostrados en la Figura 26.

El primer subproceso consiste en la aplicación de índices de la estructura del problema (cuadro 1) tomando como información de entrada el conjunto de instancias del problema a analizar y permitiendo calcular la complejidad de la estructura del problema.

El segundo subproceso consiste en la caracterización de la estructura del algoritmo (cuadro 2); a través de índices que son aplicados a la muestra del espacio de soluciones del problema, tomando como información de entrada el conjunto de instancias a analizar y el algoritmo solucionador. En ambos procesos se obtiene como salida los resultados de los índices aplicados a la estructura de la instancia y la muestra del espacio de soluciones del problema.

Y por último el tercer subproceso consiste en la caracterización del comportamiento, trayectoria (cuadro 2) y desempeño algorítmico (cuadro 3) a través de índices; recibiendo como información de entrada el conjunto de instancias a procesar y un conjunto de variantes del algoritmo solucionador, devolviendo los resultados calculados de los índices para las fases de *Identificación de las regiones de dominancia* y *Selección de los índices del problema y del algoritmo*.

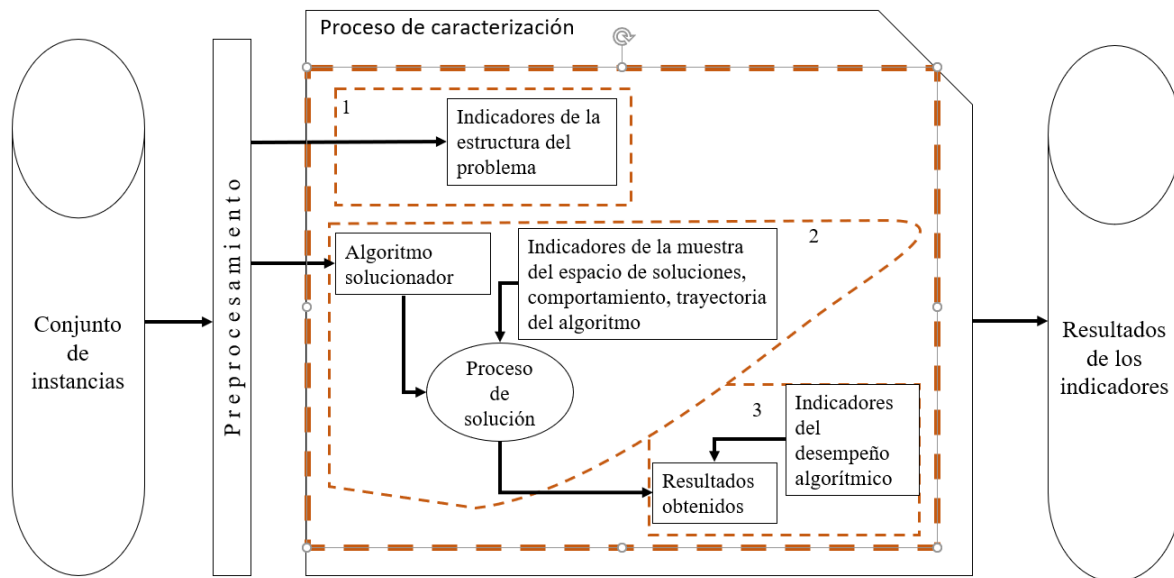


Figura 26 Proceso de Caracterización Actual en VisTHAA.

A continuación en el Algoritmo 5 se explica el procedimiento general del proceso de caracterización mostrado en la Figura 26.

Algoritmo 5 Proceso de Caracterización de Instancias y Variantes de Algoritmos Mono/Multi Objetivo.

1. **Inicio**
2. **Obtención** del conjunto de instancias a analizar → *conjuntoIns*
3. **Obtención** del conjunto de variantes del algoritmo de interés → *conjuntoAlg*
4. **Llamar** el método ProcesoCaracterización (*conjuntoIns*, *conjuntoAlg*)
5. **Llamar** el método CaracterizaciónEstructura (*conjuntoIns*)
6. **Aplicar** índices de la estructura del problema
7. **Guardar** los resultados de los índices de la estructura → *indEst*
8. **Llamar** el método CaracterizacionCompEstTrayAlg (*conjuntoIns*, *conjuntoAlg*)
9. **Aplicar** índices del comportamiento, muestra del espacio de soluciones y la trayectoria del algorítmico
10. **Guardar** resultados obtenidos de los índices del comportamiento algorítmico → *compAlg*
11. **Guardar** resultados obtenidos de los índices de la muestra del espacio de soluciones → *mueEsp*
11. **Guardar** resultados obtenidos de los índices de la trayectoria del algoritmo → *trayAlg*
13. **Guardar** resultados obtenido del algoritmo solucionador → *res*
14. **Llamar** el método CaracterizacionDesAlg (*res*)
15. **Aplicar** índices del desempeño algorítmico a los resultados obtenidos
16. **Guardar** resultados obtenidos de los índices del desempeño algorítmico
17. **Fin**

En el algoritmo 5 se inicia con la selección del conjunto de instancias y las variantes del algoritmo a analizar (Líneas 2-3), las cuales sirven como información de entrada a los procesos de caracterización mostradas en la Figura 26 (Líneas 4-16). El primer proceso en iniciar es el cálculo de la complejidad de la estructura del problema a través de la aplicación de un conjunto de índices descritos en la sección 2.4.1, devolviendo como salida los resultados calculados de los índices de la complejidad del problema (Líneas 5-7).

Después se inicia el proceso de caracterización del comportamiento, estructura y trayectoria tomando como información de entrada los resultados obtenidos en el proceso de solución del algoritmo de interés, aplicando un conjunto de índices descritos en las secciones 2.4.2 al 2.4.4, devolviendo un conjunto de los resultados calculados (Líneas 8-13).

Por último se inicia la interpretación del desempeño del conjunto de variantes del algoritmo solucionador a través de la aplicación de un conjunto de índices descritos en las secciones 2.4.5 al 2.4.6, devolviendo como resultados índices calculados (Líneas 14 - 16).

En la Tabla 15 se muestra algunos ejemplos de los índices utilizados en el proceso de caracterización tomando como información de entrada instancias pertenecientes al problema de la mochila mono objetivo y como algoritmo solucionador una metaheurística poblacional evolutiva.

Tabla 15 Ejemplos de Índices del Proceso de Caracterización.

| Tipo de índice | Índice | Problema de la mochila | | | |
|----------------------------------------------|-------------|------------------------|-----------|---------|----------|
| | | Inst15 | Inst24 | Inst100 | Inst1000 |
| Tamaño del problema | <i>p</i> | 1.87 | 2.00 | 1.28 | 2.33 |
| Relación entre pesos y tamaño del contenedor | <i>b</i> | 0.52 | 0.5 | 0.66 | 0.25 |
| | <i>t</i> | 0.12 | 0.08 | 0.01 | 0.003 |
| | <i>d</i> | 0.02 | 0.04 | 0.008 | 0.002 |
| Tendencia central de los pesos | <i>ma</i> | 95.53 | 533681.66 | 48.7 | 51.28 |
| | <i>moda</i> | 0 | 0 | 53 | 32 |
| Dispersión de pesos | <i>r</i> | 50 | 919726 | 98 | 99 |
| | <i>dm</i> | 14.83 | 252395.91 | 25.24 | 24.66 |
| | <i>s</i> | 16.62 | 294478.56 | 832.73 | 814.63 |

| | | | | | |
|---------------------------------------|----------------------------------|------------|-----------------|------------|------------|
| | <i>cv</i> | 0.17 | 0.55 | 0.59 | 0.55 |
| Forma de la distribución de los pesos | <i>amed</i> | 0.27 | 0.09 | 0.17 | 0.02 |
| | <i>amod</i> | 5.74 | 1.81 | -0.14 | 0.67 |
| | <i>curtosis</i> | 23.62 | 44.76 | 174.01 | 1818.98 |
| Muestra del espacio de soluciones | <i>vo</i> | 4119.92 | 301158751713.34 | 34777.40 | 89460.80 |
| Comportamiento del algoritmo | <i>p_f</i> | 70.06 | 69.13 | 55.56 | 81.26 |
| | <i>p_uf</i> | 29.93 | 30.86 | 44.43 | 18.73 |
| | <i>ro</i> | 0.53 | 0.33 | 0.44 | 0.35 |
| | <i>delta</i> | 2.12 | 1.49 | 1.81 | 1.54 |
| Trayectoria del algoritmo | <i>nc</i> | 0.95 | 0.97 | 0.90 | 0.98 |
| | <i>nv</i> | 1 | 1 | 1 | 1 |
| | <i>pp</i> | 0.47 | 0.48 | 0.45 | 0.49 |
| | <i>pn</i> | 0.51 | 0.50 | 0.53 | 0.49 |
| Desempeño algorítmico mono objetivo | <i>Máxima Aptitud</i> | 1453 | 13162729 | 4914 | 14635 |
| | <i>Promedio de las aptitudes</i> | 1321.38 | 11893553.1 | 4564.17 | 9666.94 |
| | <i>Radio teórico</i> | 0.99657064 | 0.98826091 | 0.99837464 | 0.99659517 |
| | <i>Porcentaje de error</i> | 0.00342936 | 0.01173909 | 0.00162536 | 0.00340483 |
| | <i>Coficiente de engaño</i> | 0.23333333 | 0.21528943 | 0.07036536 | 0.12486719 |

Una vez finalizado el proceso de caracterización, se procede con la identificación de regiones de dominancia del algoritmo solucionador.

Identificación de las Regiones de Dominancia

Esta fase tiene como objetivo la creación de un conjunto de variantes pertenecientes al algoritmo solucionador y la identificación de las variantes a través de límites impuestos por el investigador, aplicado a un conjunto de instancias de un problema; devolviendo un conjunto de variantes (regiones) en donde el desempeño obtenido es eficiente, finalizando con la obtención de regiones dominantes conformadas del conjunto de variantes del algoritmo solucionador. En el Algoritmo 6 se muestra el proceso general de la identificación de regiones de dominancia utilizada en este trabajo de tesis.

Algoritmo 6 Proceso de Identificación de Regiones de Dominancia.

1. **Inicio**
2. **Generación** de las variantes del algoritmo solucionador
3. **Ejecución** de las variantes del algoritmo solucionador
4. **Aplicación** de los índices del desempeño
5. **Obtención** de los resultados de los índices del desempeño
7. **Si** $f(x) >$ criterio de aceptación
8. Variante seleccionada
9. **caso contrario**
10. Variante no seleccionada
11. **Si** la variante domina en todas las instancias
12. Variante superior
13. **Caso contrario**
14. Variante no superior
15. **Fin**

Como se muestra en el Algoritmo 6, el proceso inicia con la creación y la ejecución del conjunto de variantes del algoritmo solucionador (Líneas 2- 3), y la aplicación de índices para cuantificar el

desempeño de cada variante del algoritmo solucionador (Línea 4), una vez obtenido el resultado de los índices se analiza para la identificación de las mejores variantes a través de criterios impuestos por el investigador, es decir criterios que permiten diferenciar las mejores variantes de las peores variantes (Líneas 7-14). Si el desempeño obtenido de la n -variante cumple con los criterios establecidos, se toma la variante como aceptada; en caso contrario no se selecciona (Líneas 7-10), por último se analiza si la variante seleccionada domina en todas las instancias se toma como variante superior, en caso contrario la variante no es superior (Líneas 11-14).

A continuación se mostrara un ejemplo completo del proceso de identificación de regiones de dominancia, tomando como ejemplo el algoritmo metaheurístico poblacional aplicado a instancias del problema de la mochila mono objetivo. En la Tabla 16 se muestra un listado de los métodos que forman cada variante del algoritmo solucionador.

Tabla 16 Variantes del Algoritmo Metaheurístico Poblacional.

| Variante | Solución Inicial | | Método de Cruza | | Método de muta | | Búsqueda Local |
|----------|------------------|------------|-----------------|-----|----------------|-----|----------------|
| | Aleatoria | Heurística | MC1 | MC2 | MM1 | MM2 | |
| V1 | X | | X | | X | | X |
| V2 | X | | X | | X | | |
| V3 | X | | X | | | X | X |
| V4 | X | | X | | | X | |
| V5 | X | | | X | X | | X |
| V6 | X | | | X | X | | |
| V7 | X | | | X | | X | X |
| V8 | X | | | X | | X | |
| V9 | | X | X | | X | | X |
| V10 | | X | X | | X | | |
| V11 | | X | X | | | X | X |
| V12 | | X | X | | | X | |
| V13 | | X | | X | X | | X |
| V14 | | X | | X | X | | |
| V15 | | X | | X | | X | X |
| V16 | | X | | X | | X | |

Donde:

- MC1= Cruza OrderCrossOver
- MC2= Cruza de dos puntos
- MM1= Muta Inserción
- MM2= Muta de dos flips

En la Tabla 16 se muestra que cada variante está formada por una combinación de métodos del algoritmo solucionador, por ejemplo, la variante 1 está formado por un método de generación poblacional a través de una heurística, cruza orderCrossOver, muta inserción y la aplicación de la búsqueda local.

Una vez identificado el conjunto de variantes, se inicia el proceso de ejecución de cada una de ellas y se obtiene el desempeño de cada una para el análisis de regiones dominantes. Una vez finalizado el proceso de ejecución se procede a la identificación de variantes dominantes del algoritmo

metaheurístico poblacional, a través de criterios establecidos por el investigador los cuales se muestran a continuación para el problema de la mochila mono objetivo:

- 1) Si el porcentaje de error del desempeño de la variante es menor al 5% se toma en consideración para la dominancia,
- 2) En caso contrario se calculará un promedio del desempeño de dicha variante ejecutando una cantidad de 30 veces y se comparará con el promedio obtenido del desempeño de la variante actual, si dicho desempeño es mayor al promedio, también tomarlo en cuenta.

En caso de que no cumplan ninguna de las condiciones, dicha variante no se tomara en cuenta para la región de dominación. Finalmente se devuelven las regiones de dominación las cuales son las mejores variantes encontradas en esta fase.

En la Tabla 17 se muestra la representación binaria de las variantes dominantes mostradas en la Tabla 16, siguiendo los criterios mencionados previamente.

Tabla 17 Representación Binaria de las Regiones de Dominancia del Algoritmo Solucionador.

| | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | v12 | v13 | v14 | v15 | v16 |
|---------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|
| Instancia15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Instancia24 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Instancia100 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Instancia1000 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 4 | 4 | 3 | 2 | 2 | 2 | 2 | 2 |

En esta fase se escogieron las variantes 1, 2, 9, 10, debido a que predominan en todas las instancias, esto quiere decir que el desempeño calculado cumple con los criterios establecidos, demostrando que el conjunto de variantes seleccionados son superiores en comparación a los demás.

Una vez finalizado el proceso de identificación de regiones dominantes, se inicia el análisis de los resultados obtenidos del proceso de caracterización para la fase *Selección de índices del problema y del algoritmo*.

Selección de los Índices del Problema y del Algoritmo

Esta fase tiene el objetivo de identificar índices relevantes del proceso de caracterización, tomando como entrada resultados obtenidos de la estructura del problema, espacio de soluciones, comportamiento y trayectoria del algoritmo, a través de un análisis general y/o visual en forma de gráfica para su comprensión, devolviendo un conjunto de índices significativos para el investigador, las cuales servirán como información de entrada para la fase *Análisis causal*.

En el Algoritmo 7 se explica el proceso general de los índices significativos utilizados en el proceso de caracterización.

Algoritmo 7 Proceso de Selección de Índices Significativos.

1. **Inicio**
2. **Obtención** de los resultados de índices de la estructura del problema → indEst
3. **Obtención** de los resultados de índices de la muestra del espacio de soluciones del problema → mueEsp
4. **Obtención** de los índices del comportamiento del algoritmo → compAlg
5. **Obtención** de los resultados de los índices de la trayectoria del algoritmo → trayAlg

6. **Aplicación** del análisis general y/o visual a indEst, mueEsp, compAlg, trayAlg
7. **Selección** de índices significativos a indEst, mueEsp, compAlg, trayAlg
8. **Fin**

Como se muestra en el Algoritmo 7, el proceso inicia con la obtención de los resultados de los índices del proceso de caracterización (Líneas 2-5), después se aplica un análisis visual de los resultados del conjunto de índices a través de gráficas, permitiendo al investigador una mejor comprensión de los valores obtenidos (Línea 6). Por último se aplica un análisis a las gráficas obtenidas, seleccionando aquellos índices con un valor significativo para el investigador (Línea 7).

Aquí se muestran ejemplos de gráficas de índices significativos utilizados en el proceso de caracterización aplicado al problema de la mochila mono objetivo y un algoritmo metaheurístico poblacional, tomando como base la variante 1.

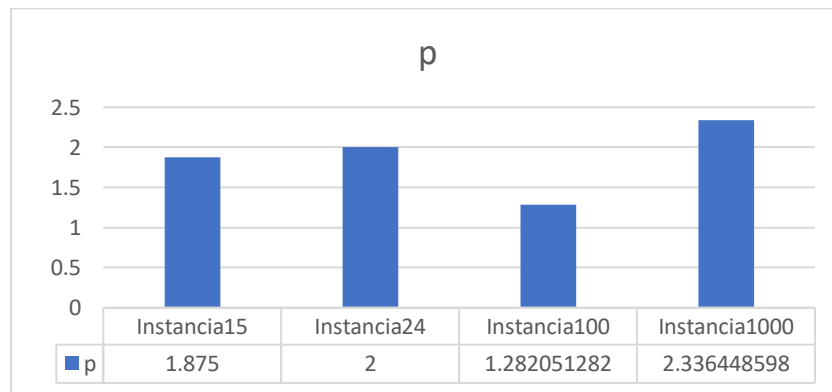


Figura 27 Índice del Tamaño del Problema de la Variante 1.

En la Figura 27 se muestra de manera visual la relación entre el número total de elementos y el número óptimo de elementos asignados al contenedor, entre más cercano al 1 indica que todos los elementos se encuentran asignados en el contenedor.

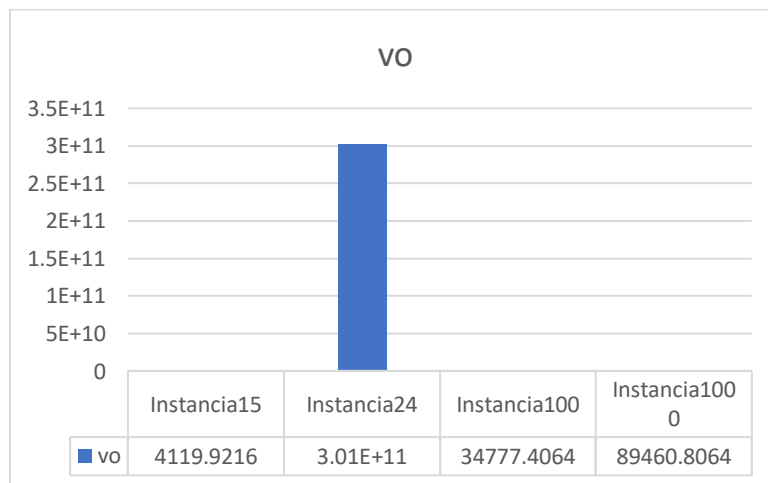


Figura 28 Índice de la Muestra del Espacio de Soluciones de la Variante 1.

En la Figura 28 se presenta de manera gráfica, la representación de una muestra del desempeño del espacio de soluciones del problema a resolver, tomando como desempeño el beneficio obtenido de cada una de las soluciones de la muestra. Para este caso la *Instancia24* obtuvo un mayor desempeño de la muestra, debido a que el beneficio de cada uno de los elementos de la instancia es mayor en comparación a las demás.

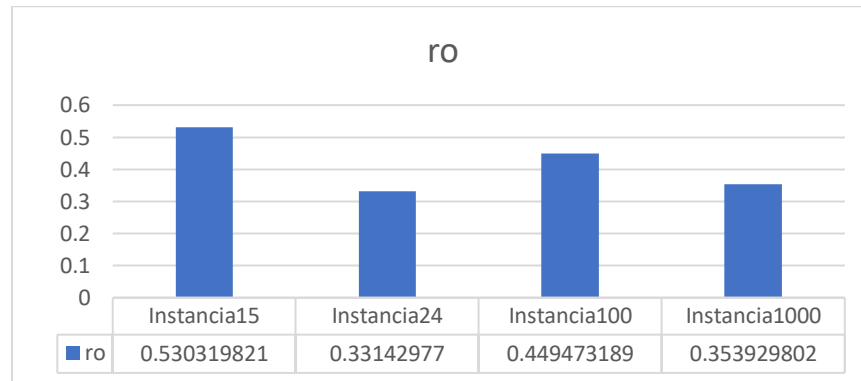


Figura 29 Índice de Comportamiento del Algoritmo de la Variante 1.

En la Figura 29 se muestra de manera gráfica el comportamiento que tiene el algoritmo en el proceso de solución, tomando como ejemplo el índice *ro* el cual, mide la rugosidad de la superficie de aptitud de un problema. Si el valor calculado es cercano al cero, indica que los resultados tienen poca correlación entre ellos, y si el valor es cercano al uno, muestra que la superficie es menos rugosa. Tomando como ejemplo la Figura 30, la *Instancia15* muestra el mejor resultado calculado indicando una superficie menos rugosa en comparación con las demás.

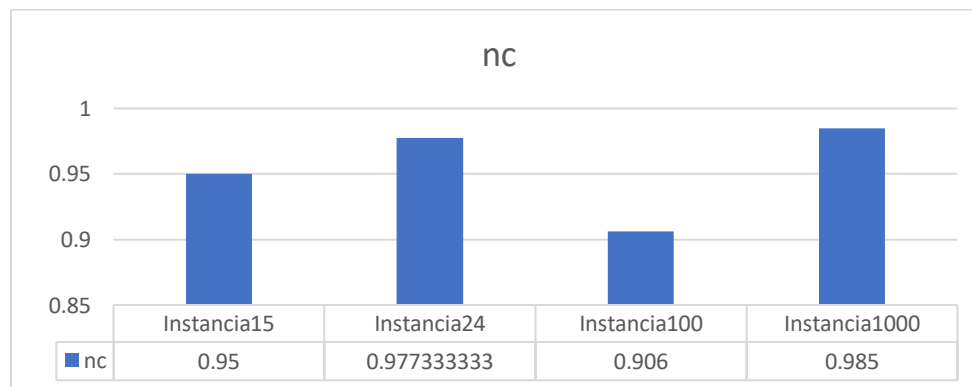


Figura 30 Índice de la Trayectoria del Algoritmo de la Variante 1.

La Figura 30 indica de manera gráfica la trayectoria del algoritmo en el proceso de solución, tomando como ejemplo el índice *nc* el cual, indica el cambio de dirección de los valores de la función aptitud, es decir si hay un cambio positivo o negativo entre dos soluciones consecutivas. Tomando como ejemplo la Figura 31, la *Instancia100* muestra mejor resultado, debido a que el promedio del número de cambios entre aptitudes fue menor en comparación con las demás.

Una vez finalizado el proceso de selección de índices significativos, se inicia la creación del modelo causal a través del algoritmo PC, tomando como información de entrada los índices seleccionados en esta fase.

Análisis Causal

Esta fase tiene como objetivo la creación del modelo causal a través del algoritmo PC (ver sección 2.3.4) para el rediseño algorítmico, tomando como información de entrada los índices significativos y las regiones dominantes obtenidas en las fases anteriores, devolviendo un grafo causal con las ecuaciones del modelo. En el Algoritmo 8 se explica el procedimiento de la generación del modelo causal.

Algoritmo 8 Proceso de Generación del Modelo Causal.

1. **Inicio**
2. **Llamar** calculoRegionesDominancia() → regionesDominantes
3. **Llamar** identificaciónIndicesSignificantes → indicesSignificantes
4. **Llamar** mapeoIndicesSignificantes(regionesDominantes, indicesSignificantes) → indicesMapeados
5. **Llamar** AnalisisCausal (indicesMapeados)
6. **Ejecución** del algoritmoPC
7. **Guardado** del modelo causal
8. **Fin**

Como se muestra el algoritmo 8 el proceso inicia con dos subprocesos: Primero.- Se identifican las regiones de dominancia, el cual consiste en la identificación de la/las mejores variantes al resolver un conjunto de instancias (Línea 2). Segundo.- Y la identificación de índices significativos a través del análisis general y/o visual (Línea 3). Una vez obtenidos ambos conjuntos, se aplica un proceso de mapeo (Línea 4), en el cual dicha información servirá como datos de entrada para el proceso de análisis causal, a través del algoritmo PC (ver sección 2.3.4), devolviendo un grafo dirigido indicando la dependencia de conjunto de índices entre si (Líneas 5-7).

A continuación se muestra un ejemplo del proceso de generación del modelo causal del algoritmo metaheurístico poblacional, aplicado a instancias del problema de la mochila mono objetivo a través de la herramienta Tetrad [Carnegie06], utilizando como información de entrada las regiones dominantes e índices significativos obtenidos de las fases: *Identificación de las regiones de dominancia* y *Selección de los índices del problema y del algoritmo*.

Para la creación del modelo causal, primero se debe de hacer un mapeo de los índices obtenidos en el proceso de selección de índices significantes, a través de las regiones de dominancia obtenidas. En la Figura 31 se muestra un ejemplo del proceso de mapeo del índice p .

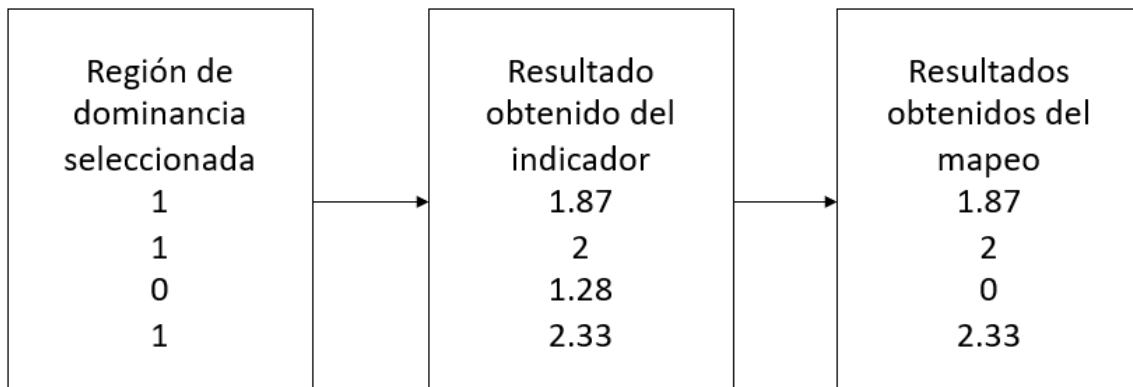


Figura 31 Ejemplo del Mapeo de Índices para el Modelo Causal.

El proceso consiste en la obtención los de los índices seleccionados en la fase *Selección de los índices del problema y del algoritmo* y el conjunto de regiones de dominancia calculados en la fase *Identificación de las regiones de dominancia* obteniendo como resultado un conjunto de índices mapeados, el cual servirá para la creación del modelo causal. En la Figura 32 se explica el proceso de creación del modelo causal utilizado en este trabajo de tesis, tomando como información de entrada las regiones de dominancia identificadas en la tercera fase (cuadro 1) y los índices significativos identificados en la cuarta fase (cuadro 2).

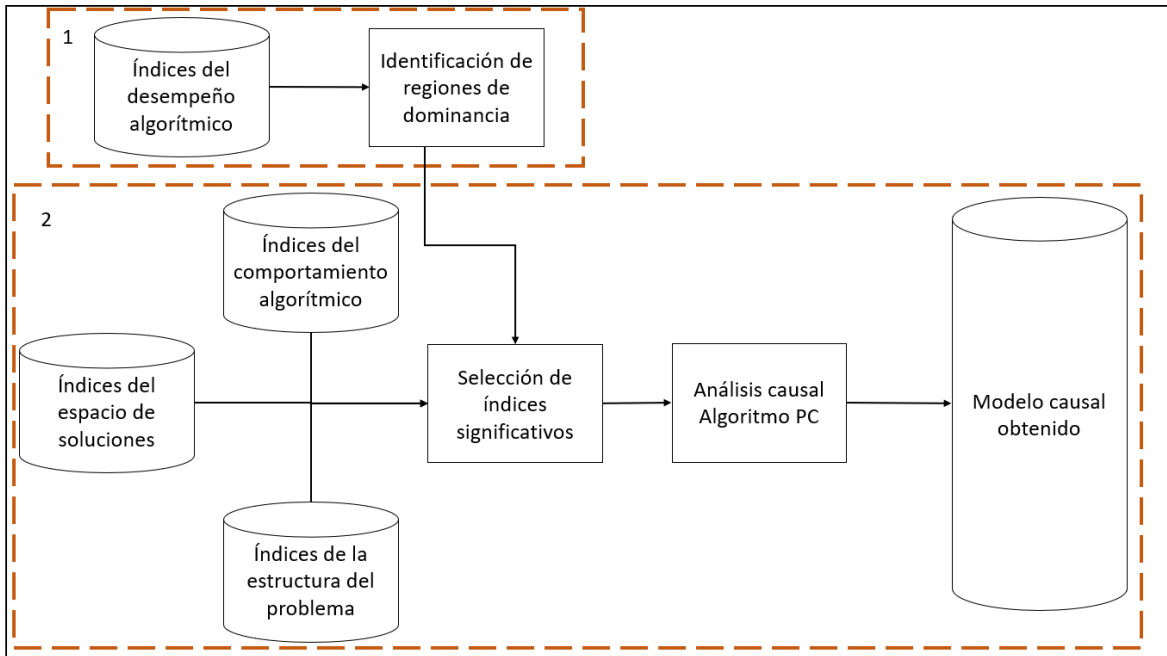


Figura 32 Proceso del Modelado Causal.

En la Figura 33 se muestra los elementos utilizados para la creación del modelo causal en Tetrad [Carnegie06] y una explicación de los elementos que conforman para la generación del modelo.

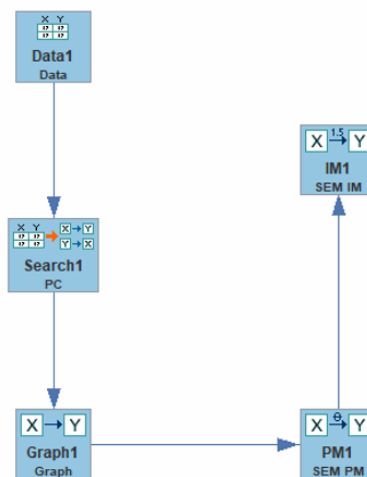


Figura 33 Elementos para la Creación del Modelo Causal en Tetrad.

1. El módulo *data* permite almacenar la información obtenida de los índices calculados en el proceso de caracterización y las regiones de dominancia obtenidos.
2. El módulo *search* permite especificar el tipo de algoritmo que se desea aplicar al conjunto de datos del módulo *data*.
3. El módulo *graph* permite crear y visualizar el grafo dirigido, obtenido del algoritmo PC.
4. El módulo *parametric model (pm)* permite generar un modelo paramétrico del grafo dirigido obtenido, permitiendo en cada vértice obtener un valor (peso) de la relación.
5. El módulo *Instantiated model (im)* permite asignar el peso de las relaciones del grafo dirigido, a través de un valor probabilístico. La suma de los valores probabilísticos de los vértices debe de dar 1.

Una vez especificado los elementos para la creación del modelo causal, se procede a ejecutar el algoritmo PC al conjunto de datos el cual está conformado por los índices del proceso de caracterización, que fueron mapeados con las regiones de dominancia, al finalizar el algoritmo PC se obtiene el grafo dirigido con las ecuaciones del modelo causal. En la Figura 35 se muestra un ejemplo del grafo obtenido del algoritmo PC aplicado al problema de la mochila mono objetivo y el algoritmo solucionador, y en la Tabla 18 se muestra el conjunto de ecuaciones obtenidas del modelo causal mostrado en la Figura 34.

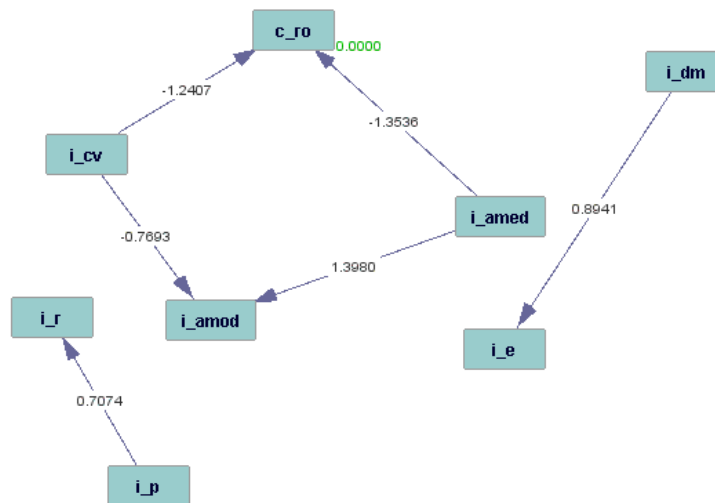


Figura 34 Ejemplo del Modelo Causal del Algoritmo Genético en Tetrad.

Tabla 18 Ecuaciones del Modelo Causal del Algoritmo Genético.

| |
|---------------------------------------------------|
| $C_{ro} = -1.2407 (i_{cv}) + 1.3536 (i_{amed})$ |
| $I_{amod} = -0.7693 (i_{cv}) + 1.3980 (i_{amed})$ |
| $I_e = 0.8941 (i_{dm})$ |
| $I_r = 0.7074 (i_p)$ |

Las ecuaciones mostradas en la Tabla 18 son aquellas en las cuales los nodos son apuntados por uno o más nodos, tomando como ejemplo los índices utilizados para el modelado causal, los cuales se interpretan que un índice es afectado por otro índice.

Una vez ejecutado el algoritmo PC, se obtiene un modelo causal, el cual representa las relaciones entre los índices de la estructura del problema, del algoritmo y las regiones de dominancia,

permitiendo una estimación de las relaciones causales entre los índices, caracterizando las relaciones de mayor importancia para dar una explicación formal del porque un algoritmo es el más adecuado para resolver un subconjunto de instancias.

Al finalizar el proceso de modelado, se inicia el proceso de ejecución del algoritmo rediseñado, tomando como guía el modelo causal sugerido.

Rediseño de la Estructura del Algoritmo

El objetivo de esta fase consiste en la aplicación del rediseño causal al conjunto de variantes que no fueron seleccionadas en la fase *Identificación de las regiones de dominancia*, tomando como base el modelo causal sugerido en la fase *Análisis causal*.

En la Tabla 19 se muestra un ejemplo de la aplicación del rediseño causal a un algoritmo metaheurístico poblacional que da solución a instancias del problema de la mochila mono objetivo.

Tabla 19 Algoritmo Metaheurístico Poblacional Rediseñado.

| Procedimiento del rediseño del algoritmo metaheurística poblacional |
|-------------------------------------------------------------------------------------------------------|
| 1. Inicio |
| 2. $b=0$; mejor beneficio |
| 3. $bc=0$; mejor de corrida |
| 4. $pc=70$; porcentaje de cruza |
| 5. $pm=5$; porcentaje de muta; |
| 6. $c=30$; número de corridas |
| 7. Lectura de instancia |
| 8. Calcular cv y $amed$ de los parámetros de la instancia del problema |
| 9. Cálculo de la media de cv y $amed$ |
| 10. Generación de la población inicial aleatoria |
| 11. Si $(cv) > media_cv$ y $(Amed) < media_amed$ entonces |
| 12. Para $i= 1$ a c |
| 13. Repetir $i=1$ a número de generaciones |
| 14. Aplicar técnica de selección de torneo determinístico |
| 15. Aplicar técnica de cruza ordercrossover |
| 16. Aplicar técnica de muta de inserción |
| 17. Aplicar Búsqueda Local |
| 18. Obtención del mejor beneficio; ba |
| 19. Si ba es mayor que b entonces |
| 20. $b=ba$ |
| 21. Fin del ciclo |
| 22. Obtener el beneficio de la corrida; bc |
| 23. Si bc es mayor a b |
| 24. $B= bc$ |
| 25. En caso contrario |
| 26. Si $pc < 100$ entonces |
| 27. $Pc= pc+10$ |
| 28. Si $pm > 1$ entonces |
| 29. $Pm=pm-1$ |
| 30. Fin del ciclo para |
| 31. En caso contrario |
| 32. Para $i= 1$ a c |
| 33. Repetir $i=1$ a número de generaciones |
| 34. Aplicar técnica de selección de torneo determinístico |
| 35. Aplicar técnica de cruza ordercrossover |
| 36. Aplicar técnica de muta heurística |
| 37. Aplicar Búsqueda Local |
| 38. Obtención del mejor beneficio; ba |
| 39. Si ba es mayor que b entonces |
| 40. $b=ba$ |

41. Fin del ciclo
42. Obtener el beneficio de la corrida; bc
43. Si bc es mayor a b
44. B= bc
45. En caso contrario
46. Si $pc < 100$ entonces
47. $Pc = pc + 10$
48. Si $pm > 1$ entonces
49. $Pm = pm - 1$
50. Fin del ciclo para
51. Fin

El proceso del rediseño algorítmico realizado a través del modelo causal sugerido por Tetrad, permitió mejorar el desempeño de todas las variantes mencionadas en la fase *Identificación de las regiones de dominancia*. En el capítulo 4 se muestran los resultados obtenidos del desempeño de cada variante en la Tabla 16, y los resultados obtenidos de la aplicación del rediseño algorítmico para cada variante. Así también como su porcentaje de mejora, comparando el desempeño de cada variante con el resultado obtenido del rediseño.

Capítulo 4: Experimentación y Análisis de Resultados

En este capítulo se muestran los resultados obtenidos de la aplicación de las metodologías y procesos propuestos en el Capítulo 3, aplicado al rediseño del módulo de preprocesamiento de instancias, la caracterización de la estructura del problema, la implementación del módulo de caracterización del desempeño final y el rediseño de algoritmos a través de técnicas causales; aplicado a los problemas de optimización mono y multi objetivo propuestos en el Capítulo 2. Como contribución se incorporaron nuevos módulos aplicados a la herramienta VisTHAA, para el aumento de su funcionalidad.

4.1 Ambiente Experimental

En esta sección se muestran las características del equipo experimental, las herramientas de software que se emplearon, las instancias analizadas y el algoritmo solucionador para el cumplimiento de este trabajo de tesis. En la Tabla 20 se muestra las características del equipo en donde se realizaron las experimentaciones y los programas utilizados para el desarrollo del preprocesamiento de instancias y el rediseño causal de algoritmos mono/multi objetivo.

Tabla 20 Características del Equipo Experimental.

| Hardware | Software |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Laptop portatil, Intel core i7-6700HQ CPU 2.60GHz, 16Gb de RAM, S.O Windows 10 Home Single Language x64 | <ul style="list-style-type: none"> • NetBeans 8.1 • Tetrad 5.0.0 |

En la Tabla 21 se describen las características del conjunto de instancias pertenecientes a los problemas analizados en el proceso de caracterización y rediseño causal.

Tabla 21 Características del Conjunto de Problemas Analizados.

| Problema | Objetivo | Número de instancias | Características | |
|---------------------------------------------------|----------|----------------------|-------------------------|---------------------------|
| Mochila | Mono | 4 | Número de objetos | 15, 24, 100, 1000 |
| | | | Capacidad de la mochila | 750, 6404180, 3254, 13001 |
| Empacado de objetos con pesos y tamaños variables | Mono | 10 | Número de objetos | 250 |
| | | | Número de contenedores | 250 |
| Selección de portafolio de cartera de proyectos | Multi | 3 | Presupuesto máximo | 250000 |
| | | | Número de objetivos | 9 |
| | | | Número de áreas | 2 |
| | | | Número de regiones | 3 |
| | | | Número de proyectos | 100 |

La Tabla 22 especifica las configuraciones del conjunto de algoritmos solucionadores utilizados en el proceso de caracterización y en el proceso de rediseño causal.

Tabla 22 Características de Conjunto de Algoritmos Solucionadores.

| Algoritmo solucionador | Parámetros | Valores |
|------------------------|------------------------|---------|
| Genético | Tamaño de la población | 100 |
| | Número de generaciones | 500 |
| | Porcentaje de cruza | 70% |
| | Porcentaje de muta | 5% |
| VNS-VCSBPP | N1(x) | 15% |
| | N2(x) | 5% |
| NSGA-II | Tamaño de la población | 100 |
| | Número de generaciones | 500 |
| | Porcentaje de cruza | 70% |
| | Porcentaje de muta | 5% |

4.2 Experimentación 1: Preprocesamiento de Instancias

Esta sección muestra los resultados obtenidos de la experimentación del rediseño del módulo de preprocesamiento de instancias de la herramienta VisTHAA, aplicado a los problemas de la mochila mono objetivo, empaçado de contenedores con pesos y tamaños variables mono objetivo y selección del portafolio de cartera de proyectos multi objetivo, los cuales pertenecen a la familia de contenedores.

La Figura 35 describe la interfaz principal de la herramienta VisTHAA, con la incorporación de los nuevos módulos propuestos en esta tesis. La funcionalidad de los módulos se explica a lo largo de este capítulo.

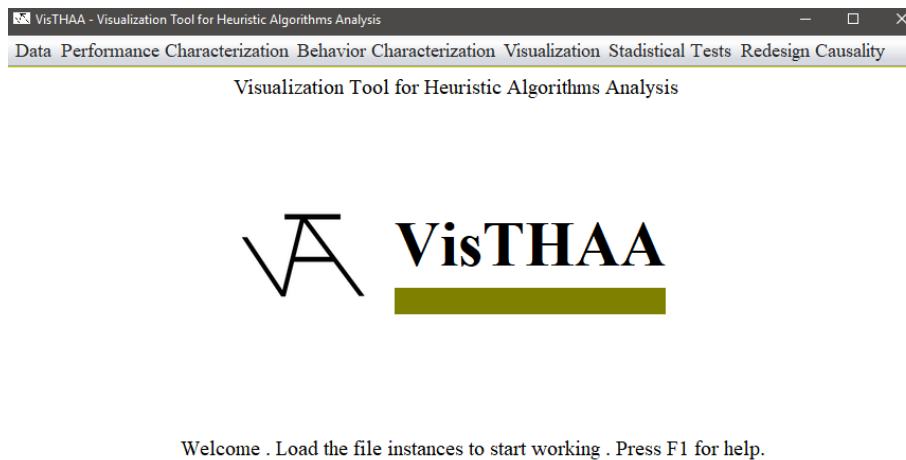


Figura 35 Ventana Principal de VisTHAA.

En este trabajo se planteó la incorporación de un módulo que facilite al investigador la selección de problema a analizar, en la Figura 36 se presenta la interfaz del módulo de *problem selection* propuesto en esta tesis. En el Algoritmo 9 se describen los pasos del funcionamiento interno del módulo mencionado.

Algoritmo 9 Proceso del Módulo *Problem Selection*.

1. **Inicio**
2. **Selección** del tipo de objetivo del problema
3. **Selección** del tipo de tipo de problema
4. **Selección** del tipo de instancia
5. **Ejecución** del preprocesamiento de instancias
6. **Guardado** del tipo de objetivo del problema → tipoObjetivo
7. **Guardado** del tipo de problema → tipoProblema
8. **Guardado** del tipo de instancia → tipoInstancia
9. **Búsqueda** de la selección del archivo logbook
10. **Si** el tipo de objetivo, problema o instancia del logbook es igual al tipoObjetivo, tipoProblema, tipoInstancia
11. **Ejecución** del método de preprocesamientoInstancias()
12. **Sino**
13. Archivo logbook no pertenece al tipo de problema a procesar
14. **Fin**

El algoritmo 9 inicia con la petición de la sección del tipo de objetivo, problema e instancia (Líneas 2-4). Una vez especificados dichos elementos se guarda cada una en variables y se inicia el preprocesamiento con la búsqueda del archivo logbook (Líneas 5-9), si el archivo seleccionado pertenece al tipo de objetivo, problema e instancia declarado anteriormente, se ejecuta el preprocesamiento del conjunto de instancias del archivo logbook seleccionado (Líneas 10-13).

En el primer panel (izquierdo) presenta un listado de opciones del tipo de problema y el tipo de instancia con el que se desea trabajar para problemas mono objetivo y el segundo panel (derecho) muestra únicamente un listado de problemas multi objetivo que dispone la herramienta.

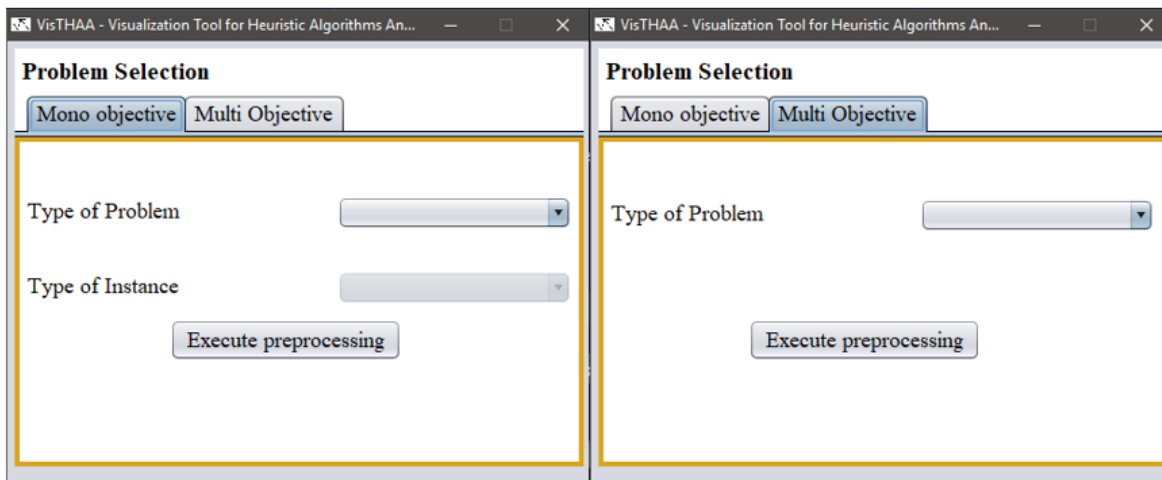


Figura 36 Módulo *Problem Selection* en VisTHAA.

Cada botón *Execute processing* mostrado en la Figura 36, tiene una función propia, el botón en el panel izquierdo permite guardar el tipo de objetivo, el tipo de problema y el tipo de instancia de los problemas mono objetivo a analizar, mientras que el botón del panel izquierdo guarda únicamente el tipo de objetivo y el tipo de problema de las instancias multi objetivo.

Una vez especificado el tipo de problema a analizar, se inicia el preprocesamiento haciendo clic en el botón *Execute processing*, se inicia el preprocesamiento con la búsqueda del archivo logbook el cual

contiene el nombre del conjunto de instancias a analizar y el archivo metainstance. En la Figura 37 se muestra la ventana de búsqueda del archivo logbook.

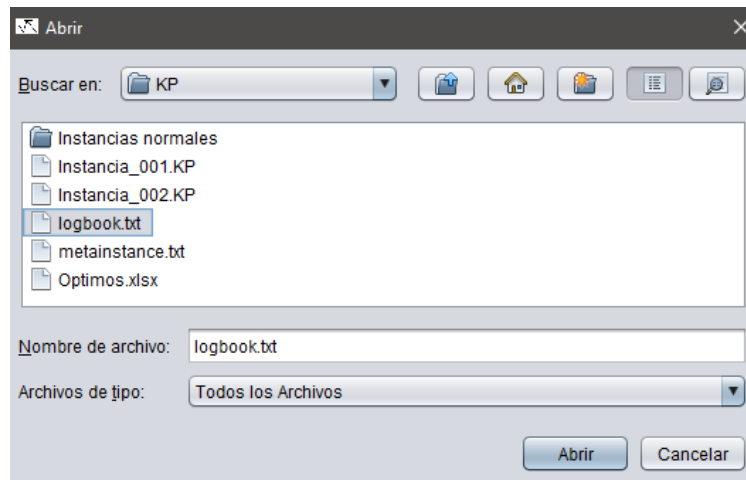


Figura 37 Ventana de Búsqueda del Archivo logbook.

Después de la selección del archivo logbook, se procesa el conjunto de instancias que contiene el archivo logbook y el archivo metainstance (Ver sección 3.1). A continuación en las Figuras 38, 39 y 40, se muestran ventanas que indican que el preprocesamiento del conjunto de instancias aplicado a los casos de estudio fueron aceptados.

4.2.1 Resultados obtenidos del Preprocesamiento de Instancias

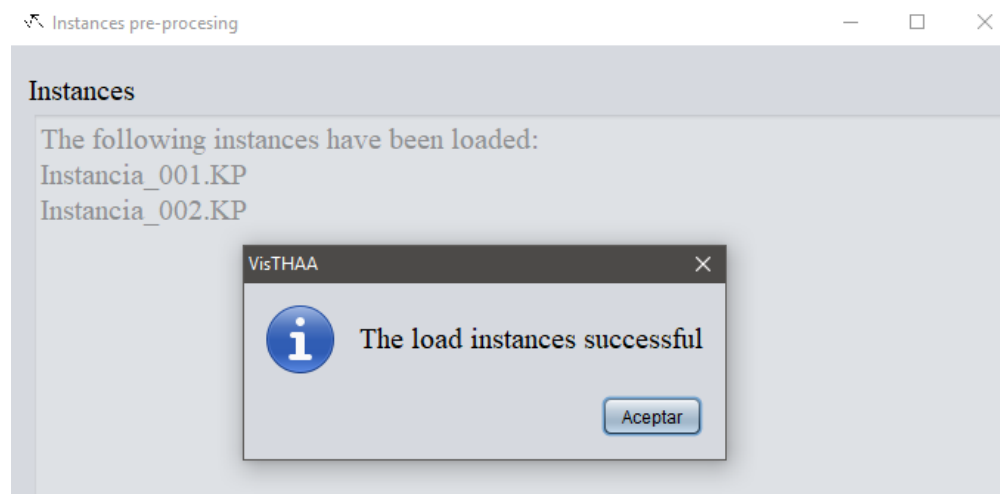


Figura 38 Ventana de Instancias Cargadas del Problema de la Mochila en VisTHAA.

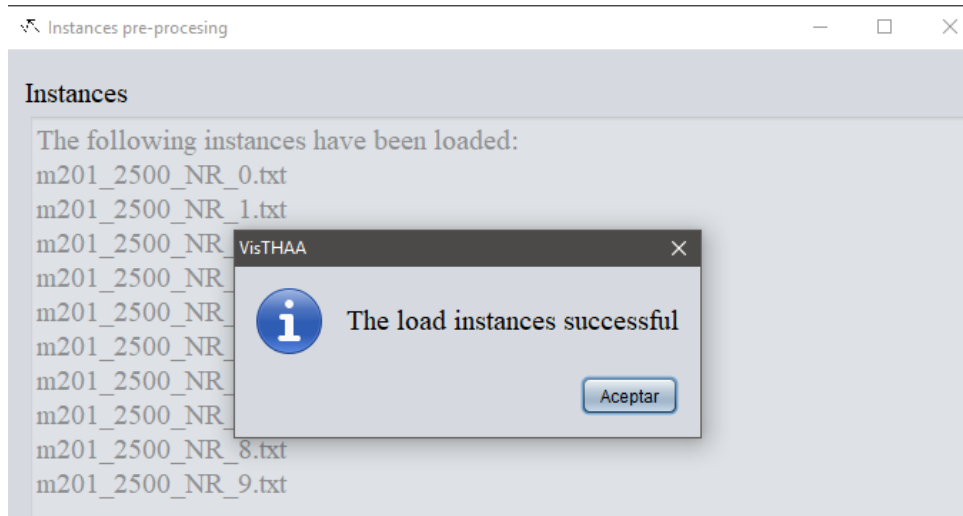


Figura 39 Ventana de Instancias Cargadas del Problema de Empacado de Contenedores con Pesos y Tamaños Variables en VisTHAA.

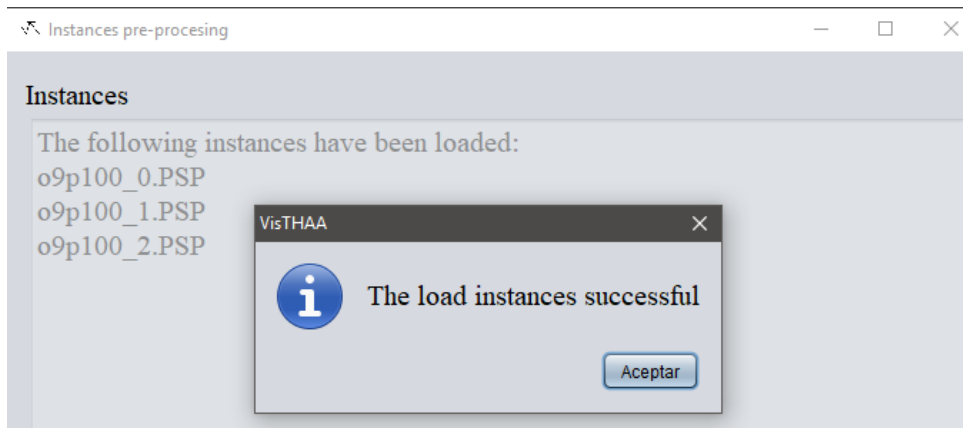


Figura 40 Ventana de Instancias Cargadas del Problema de Selección de Portafolio de Proyectos en VisTHAA.

4.3 Experimentación 2: Caracterización de la Estructura del Problema

En esta sección se visualizan los resultados obtenidos de la aplicación de la metodología diseñada en el capítulo 3, enfocándose en el proceso de caracterización de la estructura del problema. Para utilizar este proceso, se usa el módulo *Attribute Matrix*, el cual se encuentra en la siguiente ruta: *data* → *measurements* → *Attribute Matrix*

En la Figura 41 se muestra el módulo de matriz de atributos, el cual tiene como función la caracterización de la estructura del problema a través de índices descritos en el capítulo 2, el cual está formado por los siguientes elementos: un panel (izquierdo) para mostrar las instancias cargadas, un panel (derecho) el cual contiene índices de la estructura del problema y el botón *calculate* que se encarga de dar inicio del proceso de cálculo de los índices.

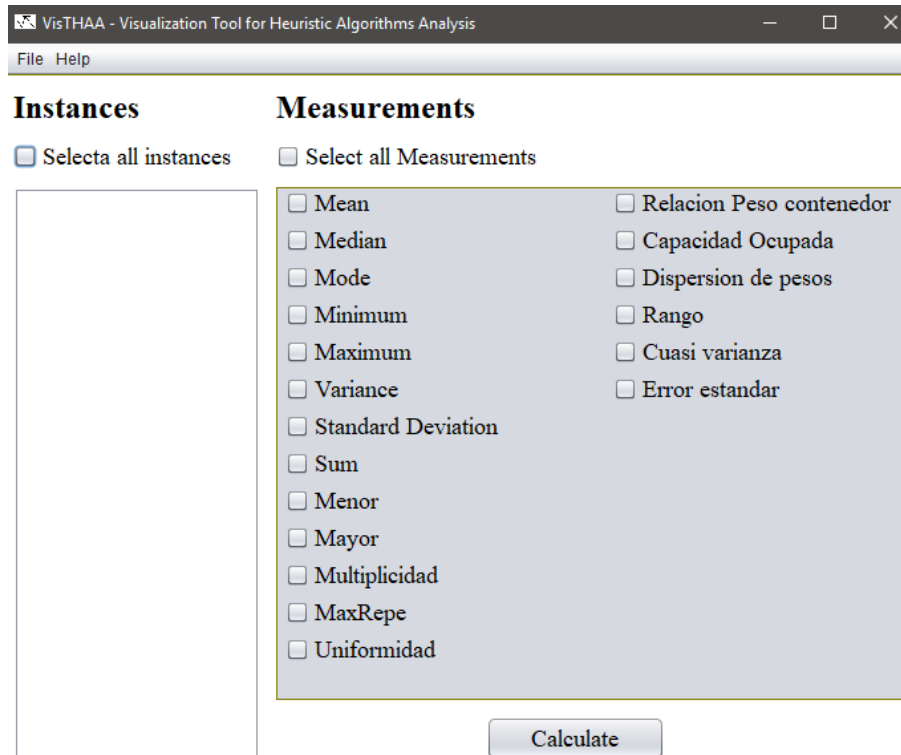


Figura 41 Módulo de Attribute Matrix.

El Algoritmo 10 describe el seguimiento de las funciones del módulo *Attribute Matrix*.

Algoritmo 10 Proceso del Módulo *Attribute Matrix*.

1. **Inicio.**
2. **Selección** de las instancias a caracterizar
3. **Selección** del conjunto de índices a utilizar
4. **Llamar** el proceso de caracterización del módulo *Attribute Matrix*
5. **Guardar** el número de instancias a caracterizar $\rightarrow numInstances$
6. Ciclo $i=0$ a $numInstances$
7. **Obtención** del peso de la instancia $_i$
8. **Si** el índice esta seleccionado
9. **Aplicar** índice al peso de los objetos de la instancia seleccionada
10. **Fin** condición si
11. **Fin** ciclo i
12. **Fin**

Como se muestra en el Algoritmo 10, para dar uso del módulo; primero se debe de seleccionar las instancias que se desean caracterizar (Línea 2), y los índices que se desean aplicar al conjunto de instancias seleccionadas (Línea 3), después se inicia el proceso de caracterización aplicando al conjunto de instancias (Línea 4) a través del botón *calculate*. Por último se toma la información guardada en el preprocesamiento de instancias y se almacena en una estructura y se aplican las fórmulas de los índice seleccionados por el investigador (Líneas 5-11).

Una vez terminado el proceso de caracterización, la herramienta guarda la información en un archivo texto plano (.txt) y además se visualiza en una nueva ventana, una matriz conformada por los resultados de las instancias seleccionadas (filas) y los índices seleccionados (columnas).

4.3.1 Resultados obtenidos del Proceso de Caracterización de la Estructura del Problema

En la Tabla 23 y 24 se exponen los resultados obtenidos del proceso de caracterización de la estructura, aplicados a instancias de los problemas de empaqueo de contenedores con pesos y tamaños variables mono objetivo y selección de portafolio de proyectos multi objetivo, además se dará una explicación de los resultados obtenidos.

Tabla 23 Matriz de Atributos del Problema de Empaqueo de Contenedores con Pesos y Tamaños Variables.

| Instancias | Mean | Median | Mode | Uniformidad | Rango |
|--------------------|---------|---------|---------|-------------|----------|
| m201_2500_NR_0.txt | 794.229 | 522.000 | 48.000 | 0.465 | 2202.000 |
| m201_2500_NR_1.txt | 796.816 | 459.000 | 40.000 | 0.405 | 2208.000 |
| m201_2500_NR_2.txt | 662.289 | 388.000 | 76.000 | 0.415 | 1824.000 |
| m201_2500_NR_3.txt | 794.229 | 524.000 | 48.000 | 0.455 | 2198.000 |
| m201_2500_NR_4.txt | 653.234 | 359.000 | 60.000 | 0.386 | 1810.000 |
| m201_2500_NR_5.txt | 794.229 | 460.000 | 88.000 | 0.415 | 2200.000 |
| m201_2500_NR_6.txt | 653.234 | 405.000 | 52.000 | 0.460 | 1786.000 |
| m201_2500_NR_7.txt | 799.403 | 495.000 | 204.000 | 0.455 | 2190.000 |
| m201_2500_NR_8.txt | 794.229 | 540.000 | 48.000 | 0.480 | 2190.000 |
| m201_2500_NR_9.txt | 796.816 | 473.000 | 72.000 | 0.450 | 2186.000 |

Los resultados obtenidos del proceso de caracterización de la estructura del problema mostrados en la Tabla 23, indican que hay más objetos con un peso menor, debido a que el promedio del conjunto de ellos es mayor al obtenido en el índice *mode*. También se evidencia que el conjunto de pesos no muestran una uniformidad, debido a que el valor calculado está alejado del uno, esto quiere decir que se observan mayores diferencias de la frecuencia de aparición de cada uno de los objetos. Además el índice *mean* muestra el promedio del conjunto de pesos de los objetos, permitiendo al investigador una mejor perspectiva de la relación peso/capacidad del problema a analizar. En el caso de los índices *Median* y *Rango* no aportan información relevante que permita al investigador una mejor comprensión de la estructura del problema.

Tabla 24 Matriz de Atributos del Problema de Selección de Portafolio de Proyectos.

| Instancias | Mean | Multiplicidad | MaxRepe | Uniformidad | Rango | Error estándar |
|--------------|--------|--------------------|---------|-------------|--------|--------------------|
| o9p100_0.PSP | 7563.8 | 1.0638297872340425 | 2.0 | 0.8 | 4905.0 | 149.55592800019656 |
| o9p100_1.PSP | 7515.3 | 1.0416666666666667 | 2.0 | 0.84 | 4845.0 | 153.31607580420274 |
| o9p100_2.PSP | 7505.8 | 1.098901098901099 | 3.0 | 0.86 | 4980.0 | 146.78275988684766 |

Los resultados obtenidos mostrados en la Tabla 24, indican una frecuencia máxima con la que se repite un peso en el conjunto de objetos en el índice *maxRepe*. El índice *uniformidad* muestra la distribución uniforme del conjunto de pesos de una instancia, indicando una distribución semejante, debido a que el valor obtenido es cercano al uno. Además el índice *mean* muestra el promedio del peso del conjunto de objetos, permitiendo al investigador una comprensión de la relación peso/contenedor. Por último el índice *error estándar* mide la desviación de los valores estadísticos obtenido de un conjunto de muestras aleatorias de una población. Para el caso del índice *Rango*, de igual manera que en la Tabla 23, no aporta información relevante para el investigador.

4.4 Experimentación 3: Caracterización del Comportamiento del Algoritmo

En esta sección se presentan los resultados obtenidos de la aplicación de la metodología diseñada en el capítulo 3, enfocándose en el proceso de caracterización del comportamiento del algoritmo a través de la incorporación de un nuevo módulo mostrado en la Figura 42, el cual fue propuesto en este trabajo, dicho módulo se encuentra en la siguiente ruta: *Behavior Characterization* → *Behavior Mono/Multi Objective*.

En la Figura 42 se muestra el módulo *caracterización del comportamiento del algoritmo*, que tiene como objetivo el cuantificar el comportamiento de un algoritmo al solucionar un conjunto de instancias a través de un conjunto de índices descritos en el capítulo 2, dicho módulo está formado por los siguientes elementos: un panel (izquierdo) que muestra un listado de las instancias cargadas de la herramienta, un panel (derecho) que contiene el número de ejecuciones que el investigador desea, índices del proceso de caracterización del comportamiento, y un botón que tiene como función el inicial el proceso de cálculo de índices.

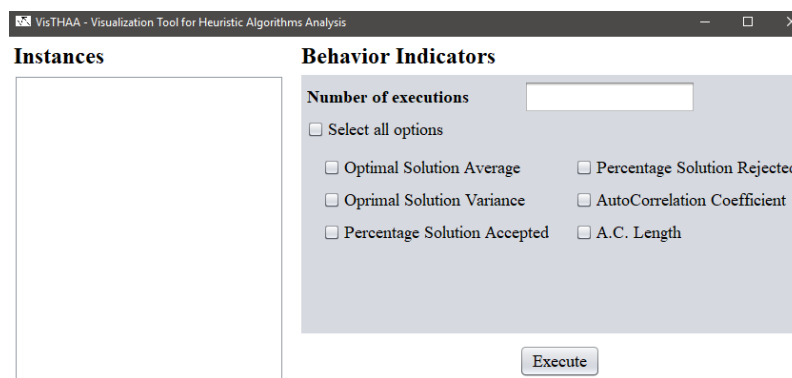


Figura 42 Módulo *Behavior Mono/Muti objective*.

El Algoritmo 11 describe el proceso interno del módulo *Behavior Mono/Muti objective* propuesto en este trabajo de tesis.

Algoritmo 11 Proceso del Módulo *Behavior Mono/Muti objective*.

1. **Inicio**
2. **Selección** del conjunto de instancias a caracterizar
3. **Selección** del conjunto de índices a utilizar
4. **Especificar** el número de ejecuciones
5. **Llamar** el proceso de caracterización del módulo Behavior Mono/Muti objective
6. **Guardar** el número de ejecuciones → *numEjecuciones*
7. **Guardar** el número de instancias a caracterizar → *numInstancias*
8. **Ciclo** $i=0$ a *numInstancias*
9. *resultadosDesempeño* [*numEjecuciones*]
10. **Ciclo** $j=0$ a *numEjecuciones*
11. **Obtención** del desempeño
12. **Guardar** el desempeño → *resultadosDesempeño[j]*
13. **Fin** ciclo j
14. **Si** el índice esta seleccionado
15. **Aplicar** índice a *resultadosDesempeño* → *resultatdoIndice*
16. **Fin** condición si
17. **Fin** ciclo i
18. **Fin**.

El Algoritmo 11 inicia con la selección de instancias, la especificación del número de ejecuciones que el investigador desea y el conjunto de índices (Líneas 2 - 4), una vez especificados dichos elementos, se guarda cada uno de ellos y se llama el algoritmo solucionador (Líneas 5 - 7), en este proceso el algoritmo se ejecuta la cantidad de veces que el investigador especificó y en cada iteración se ejecuta los índices seleccionados y se guarda los resultados alcanzados para la obtención de un valor promedio (Líneas 8 – 17).

Una vez terminada la ejecución del algoritmo, VisTHAA guarda los resultados obtenidos de los índices seleccionados a un archivo de texto plano (.txt), el cual servirá como información de entrada para el módulo de visualización gráfica.

4.4.1 Resultados obtenidos del Proceso de Caracterización del Comportamiento del Algoritmo

En la Tabla 25 y 26 se exponen los resultados obtenidos del proceso de caracterización del comportamiento del algoritmo, aplicados al VNS-VCSBPP que da solución a instancias del problema de empaqueo de contenedores con pesos y tamaños variables y el NSGA-II aplicado a instancias del problema de selección de cartera de proyectos, incluyendo una explicación de los resultados alcanzados del comportamiento algorítmico.

Tabla 25 Resultados del Proceso de Caracterización del Comportamiento del VNS-VCBPP.

| Instance | Optimal solution average | Optimal solution variance |
|--------------------|--------------------------|---------------------------|
| m201_2500_NR_0.txt | 323763.34905195347 | 0.0 |
| m201_2500_NR_1.txt | 324955.25462873664 | 0.0 |
| m201_2500_NR_2.txt | 296863.0049397755 | 0.0 |
| m201_2500_NR_3.txt | 323763.34905195347 | 0.0 |
| m201_2500_NR_4.txt | 294224.54303745145 | 0.0 |
| m201_2500_NR_5.txt | 321771.4932602417 | 0.0 |
| m201_2500_NR_6.txt | 294224.54303745145 | 0.0 |
| m201_2500_NR_7.txt | 323484.0185121378 | 0.0 |
| m201_2500_NR_8.txt | 325755.20484366524 | 0.0 |
| m201_2500_NR_9.txt | 322960.15740967036 | 0.0 |

Como conclusión del proceso de caracterización del comportamiento del algoritmo VNS-VCSBPP, se describe que el índice *Optimal solution variance* no es un índice relevante para el cálculo, debido a que no proporciona ninguna información de la varianza de los resultados, es decir, que los valores obtenidos en cada iteración son los mismos. Y el índice *Optimal solution average* muestra el promedio de los valores en cada iteración, debido a los resultados obtenidos del índice *Optimal solution variance*, se muestra que el valor calculado en cada iteración es el mismo. Dichos valores no tendrán un cambio aun si el investigador aumenta el número de iteraciones debido a las limitantes del algoritmo.

Tabla 26 Resultados del Proceso de Caracterización del Comportamiento del Algoritmo NSGA-II.

| Instance | Percentage solution accepted | Percentage solution rejected | Autocorrelation coefficient |
|--------------|------------------------------|------------------------------|-----------------------------|
| o9p100_0.PSP | 1.0 | 0.0 | 0.8991475967162832 |
| o9p100_1.PSP | 1.0 | 0.0 | 0.8956122880178089 |
| o9p100_2.PSP | 1.0 | 0.0 | 0.8993212570638941 |

Como conclusión el proceso de caracterización del comportamiento del algoritmo NSGA-II, muestra que los índices *Percentage solution accepted* y *Autocorrelation coefficient* presentan información relevante. El índice *Percentage solution rejected* no muestra información, debido a que el algoritmo NSGA-II no genera soluciones infactibles, por ende, el índice *Percentage solution accepted* muestra un valor de 1.0, esto quiere decir que todas las soluciones generadas del algoritmo son factibles. Por último el índice *Autocorrelation coefficient* indica que la muestra de la superficie de aptitudes por instancia es poco rugosa debido a que los resultados obtenidos son cercanos al 1.

4.5 Experimentación 4: Caracterización del Desempeño Algorítmico

En este punto muestran los resultados obtenidos de la aplicación de la metodología diseñada en el capítulo 3, enfocándose en el proceso de caracterización del desempeño algorítmico a través de la incorporación de un nuevo módulo mostrado en la Figura 43 y 44, los cuales fueron propuestos en este trabajo, dichos módulos se encuentran en la siguiente ruta: *Performance Characterization* → *Performance Mono Objective* o *Performance Multi Objective* dependiendo si se quiere caracterizar desempeño mono o multi objetivo.

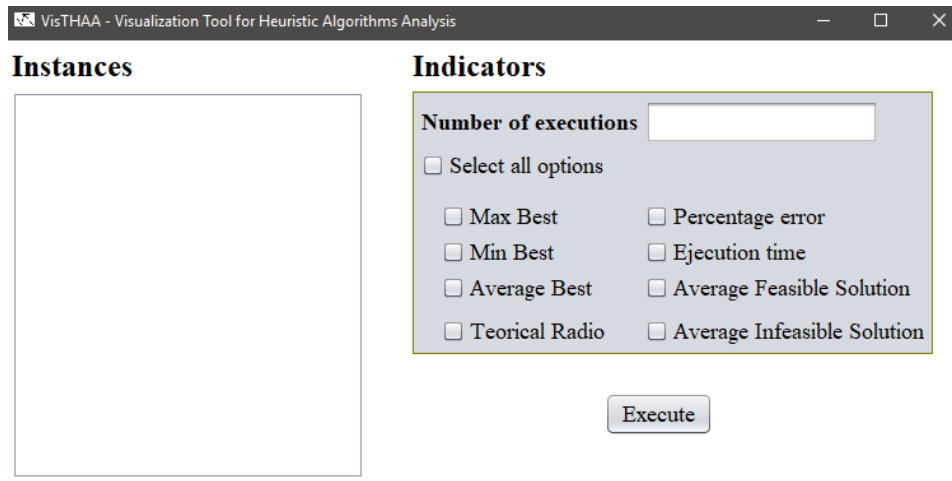


Figura 43 Módulo *Performance Mono Objective*.

En la Figura 43 el módulo del desempeño mono objetivo está conformado por los siguientes elementos: el primer panel (izquierdo) muestra las instancias del problema a analizar, el segundo panel (derecho) contiene índices del comportamiento del algoritmo y por último el botón *Execute* que da inicio el proceso de caracterización. En el Algoritmo 12 se describe el proceso interno del módulo *Performance Mono Objective* implementado en este trabajo.

Algoritmo 12 Proceso de Caracterización del Desempeño Algorítmico Mono Objetivo.

1. **Inicio**
2. **Selección** del conjunto de instancias a caracterizar
3. **Selección** del conjunto de índices a utilizar
4. **Especificar** el número de ejecuciones
5. **Llamar** el proceso de caracterización del módulo *Performance Mono Objective*
6. **Guardar** el número de ejecuciones → *numEjecuciones*
7. **Guardar** el número de instancias a caracterizar → *numInstancias*
8. **Ciclo** $i=0$ a *numInstancias*
9. resultadosDesempeño [*numEjecuciones*]
10. **Ciclo** $j=0$ a **numEjecuciones**
11. Obtención del desempeño
12. Guardar el desempeño → resultadosDesempeño[*j*]

13. **Fin** ciclo j
14. **Si** el índice esta seleccionado
15. **Aplicar** índice a resultadosDesempeño \rightarrow *resulatdoIndice*
16. **Fin** condición si
17. **Fin** ciclo i
18. **Fin.**

El Algoritmo 12 inicia con la selección del conjunto de instancias que se desean caracterizar, el número de ejecuciones que el investigador desea y la selección del conjunto de índices que se aplicaran al algoritmo (Líneas 2-4). Una vez especificadas las instancias con las que se desea analizar, el número de iteraciones y los índices a aplicar al algoritmo, se guarda cada atributo en variables para su uso posterior y se inicia el proceso de caracterización (Línea 5-7).

El módulo trabaja de la siguiente manera, se ejecuta el algoritmo n cantidad de veces por instancias y en cada iteración se aplica la fórmula de los índices seleccionados y se guarda cada resultado para obtener un valor promedio por índice seleccionado (Líneas 8-16).

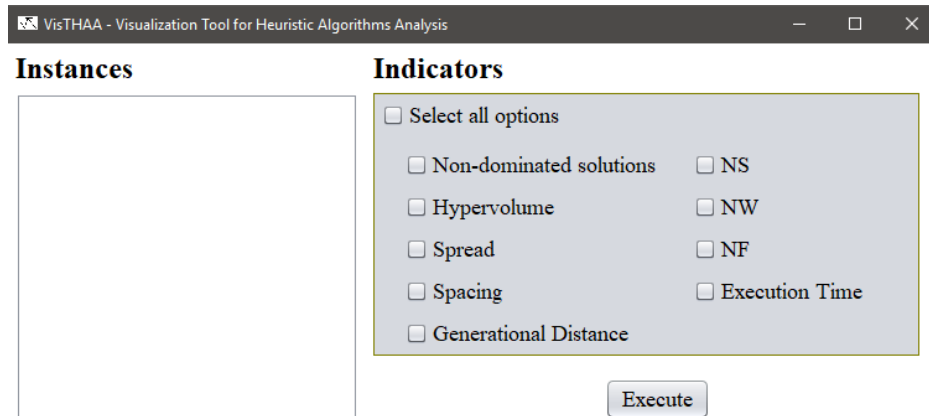


Figura 44 Módulo *Performance Multi Objective*.

Como se muestra en la Figura 44 el módulo de caracterización del desempeño algorítmico multi objetivo, está compuesto por los siguientes elementos: un panel (izquierdo) que muestra las instancias cargadas, un panel (derecho) que contiene los índices de caracterización y por último el botón *Execute* que da inicio al proceso de caracterización. En el Algoritmo 13 se describe el proceso interno del módulo *Performance Multi Objective* implementado en este trabajo.

Algoritmo 13 Proceso de Caracterización del Desempeño Algorítmico Multi Objetivo.

1. **Inicio**
2. **Selección** del conjunto de instancias a caracterizar
3. **Selección** del conjunto de índices a utilizar
4. **Llamar** el proceso de caracterización del módulo Performance Multi Objective
5. **Guardar** el número de instancias a caracterizar \rightarrow *numInstancias*
6. **Ciclo** $i=0$ a *numInstancias*
7. **Obtención** del desempeño
8. **Si** el índice esta seleccionado
9. **Aplicar** índice y guardar su resultado
10. **Fin** condición si
11. **Fin** ciclo i
12. **Fin**

El Algoritmo 13 inicia con la selección del conjunto de instancia que se desean caracterizar y el conjunto de índices que el investigador desea aplicar al desempeño del algoritmo (Líneas 2-3), después se inicia el proceso con la ejecución del algoritmo por cada una de las instancias seleccionadas obteniendo su desempeño y por último se aplican los índices seleccionados al resultado obtenido del algoritmo (Líneas 4-11)

Una vez terminado el proceso de caracterización mono o multi objetivo, la herramienta VisTHAA guarda los resultados obtenidos de los índices seleccionados a un archivo de texto plano (.txt), el cual servirá como información de entrada para el módulo de visualización gráfica.

4.5.1 Resultados obtenidos del Proceso de Caracterización del Desempeño Algorítmico

En la Tabla 27 y 28 se muestran los resultados obtenidos del proceso de caracterización del desempeño de algoritmos aplicados a los algoritmos VNS-VCSBPP que da solución a instancias del problema de empaqueo de contenedores con pesos y tamaños variables y el algoritmo NSGA-II aplicado a instancias del problema de selección de cartera de proyectos, incluyendo una explicación de los resultados obtenidos.

Tabla 27 Resultados de la Caracterización del Desempeño del Algoritmo VNS-VCSBPP.

| Instance | Max Best | Average Best | Time Execution |
|--------------------|--------------------|--------------------|----------------|
| m201_2500_NR_0.txt | 323763.34905195347 | 323763.34905195347 | 200055.0 |
| m201_2500_NR_1.txt | 324955.25462873664 | 324955.25462873664 | 200038.0 |
| m201_2500_NR_2.txt | 296863.0049397755 | 296863.0049397755 | 200034.0 |
| m201_2500_NR_3.txt | 323763.34905195347 | 323763.34905195347 | 200037.0 |
| m201_2500_NR_4.txt | 294224.54303745145 | 294224.54303745145 | 200036.0 |
| m201_2500_NR_5.txt | 321771.4932602417 | 321771.4932602417 | 200042.0 |
| m201_2500_NR_6.txt | 294224.54303745145 | 294224.54303745145 | 200045.0 |
| m201_2500_NR_7.txt | 323484.0185121378 | 323484.0185121378 | 200038.0 |
| m201_2500_NR_8.txt | 325755.20484366524 | 325755.20484366524 | 200040.0 |
| m201_2500_NR_9.txt | 322960.15740967036 | 322960.15740967036 | 200047.0 |

Como conclusión del proceso de caracterización del desempeño del algoritmo VNS-VCSBPP, muestra que el valor calculado por iteración es el mismo, debido al resultado calculado en el índice *Average Best* es igual al resultado de *Max Best*. Los resultados calculados del índice *Time Execution*, muestran que el tiempo de solución del algoritmo en cada instancia es similar.

Tabla 28 Resultados de la Caracterización del Desempeño del Algoritmo NSGA-II.

| Instance | Non dominated solutions | Hypervolume | Spread | Time Execution |
|--------------|-------------------------|-----------------------|---------------------|----------------|
| o9p100_0.psp | 0.45 | 5.134674880744137e11 | 0.21239157337725284 | 0.999 |
| o9p100_1.psp | 0.57 | 5.1289424726815015e11 | 0.27825948599472916 | 0.912 |
| o9p100_2.psp | 0.39 | 5.1290805815855676e11 | 0.25377235362229356 | 0.731 |

En conclusión el proceso de caracterización del desempeño del algoritmo NSGA-II, muestra que los resultados calculados de los índices *Hypervolume* y *Spread* obtienen resultados satisfactorios y el índice *Non dominated solutions* muestra un desempeño bajo debido a que todos los valores calculados

en todas las instancias fueron menores al .60. El resultado obtenido del índice spread muestra una buena e ideal dispersión entre las soluciones del frente de Pareto por su valor cercano al cero. Por último el índice *hypervolume* indica un buen resultado debido al amplio espacio con respecto a las soluciones pertenecientes al frente de Pareto.

4.6 Experimentación 5: Rediseño Causal de Algoritmos Mono/Multi Objetivo

En esta sección se visualizan los resultados obtenidos de cada una de las fases de la metodología del rediseño causal descrito en el capítulo 3. Dicha metodología se aplicó al problema de selección de portafolio de proyectos multi objetivo y el algoritmo NSGA-II. En la Figura 45 se muestra la interfaz del módulo de rediseño algorítmico incorporado en VisTHAA, el cual está compuesto por dos paneles, el panel izquierdo contiene el conjunto de instancias a analizar dependiendo al problema, y el panel derecho muestra el conjunto de algoritmo solucionadores.

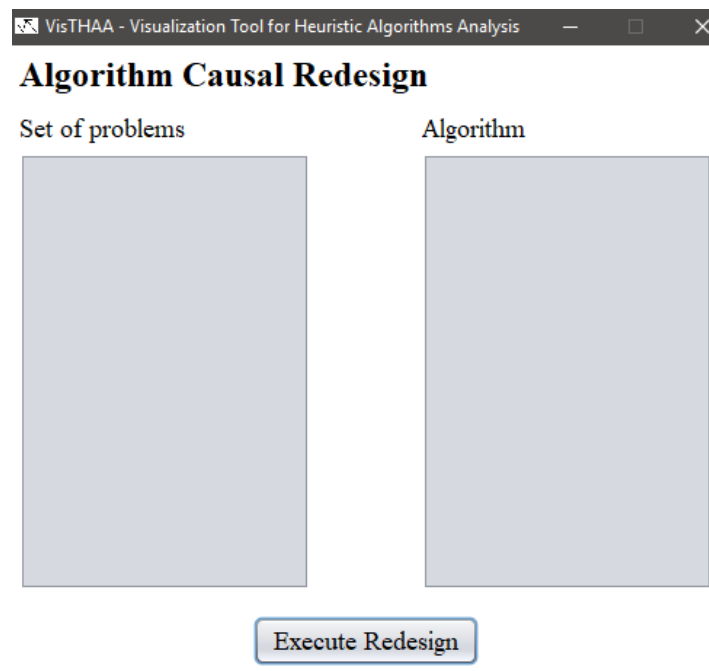


Figura 45 Módulo Algorithm Causal Redesign.

En el Algoritmo 14 se muestra el proceso interno del módulo *Algorithm Causal Redesign* implementado en este trabajo.

Algoritmo 14 Proceso del Módulo *Algorithm Causal Redesign*.

1. **Inicio**
2. **Selección** del conjunto de problemas a analizar → *instancias*
3. **Selección** del conjunto de algoritmos a utilizar → *algoritmos*
4. **Ejecución** del rediseño causal
5. **Llamar** caracterizaciónEstructuraProblema(*instancias*)
6. **Obtención** de los datos de la instancia a caracterizar → *datos*
7. **Aplicar** índices de la estructura del problema
8. **Guardar** resultados de los índices de la estructura del problema → *indiceEstructura*
9. **Llamar** caracterizaciónMuestraEspacioSoluciones(*instancias, algoritmos*)
10. **Obtención** del conjunto de datos del desempeño
11. **Aplicar** índices de la muestra del espacio de soluciones

12. **Guardar** resultados de los índices de la muestra del espacio de soluciones del problema → *indiceMuestra*
13. **Llamar** GeneraciónVariantesAlgoritmo (instancias, algoritmos) → *numCombinaciones, algoritmoVariante*
14. Ciclo $i=0$ a *numCombinaciones*
15. **Llamar** caracterizacionComportamientoAlgoritmo (*instancias, algoritmoVariante*)
16. **Obtención** del desempeño de la variante i
17. **Aplicar** índices del comportamiento del algoritmo
18. **Guardar** resultados de los índices del comportamiento de la variante i → *indiceComportamiento*
19. **Llamar** CaracterizacionTrayectoriaAlgoritmo (*instancias, algoritmoVariante*)
20. **Obtención** del desempeño de la variante i
21. **Aplicar** índices de la trayectoria del algoritmo
22. **Guardar** resultados de los índices de la trayectoria de la variante i → *indiceTrayectoria*
23. **Llamar** CaracterizacionDesempeñoAlgoritmo (*instancias,algoritmoVariante*)
24. **Obtención** del desempeño de la variante i
25. **Aplicar** índices del desempeño algoritmo
26. **Guardar** resultados de los índices del desempeño de la variante i → *indiceDesempeño*
27. **Llamar** IdentificacionRegionesDominancia (*indiceDesempeño, numCombinaciones*)
28. **Aplicación** del proceso de identificación de regiones
29. **Guardar** la posición de las mejores variantes → *regionesDominantes*
30. **Llamar** SeleccionIndicesSignificativos (*indiceEstructura, indiceMuestra, indiceComportamiento, indiceTrayectoria*)
31. **Aplicar** análisis general y/o visual
32. **Guardar** índices significativos → *indiceSignificativo*
33. **Llamar** GeneracionModeloCausal (*indiceSignificativo,regionesDominantes*)
34. **Aplicación** del algoritmo PC (*indiceSignificativo,regionesDominantes*)
35. **Guardado** del modelo causal sugerido
36. **Llamar** RediseñoAlgoritmico()
37. **Fin**

4.6.1 Resultados obtenidos del Proceso de Caracterización

En esta etapa se describen los resultados obtenidos del proceso de caracterización aplicado a instancias del problema de selección de portafolio de proyectos (ver Tabla 21), utilizando el algoritmo propuesto en el capítulo 3 (ver Algoritmo 5). En la Tabla 29 se muestra los resultados obtenidos de la aplicación de índices del proceso de caracterización descritos en las secciones 2.4.1 al 2.4.6, aplicado a la variante 1.

Tabla 29 Resultados obtenidos de los Índices del Proceso de Caracterización.

| Tipo de índice | Índice | Problema de selección de cartera de proyectos | | |
|----------------------------------------------|------------|-----------------------------------------------|---------|---------|
| | | o9p100 | o9p100 | o9p100 |
| Relación entre pesos y tamaño del contenedor | b | 0.3305 | 0.3326 | 0.3330 |
| | t | 0.0302 | 0.0300 | 0.0300 |
| | d | 0.0302 | 0.0300 | 0.0300 |
| Tendencia central de los pesos | ma | 7563.8 | 7515.3 | 7505.8 |
| | $moda$ | 8010 | 5410 | 8435 |
| Dispersión de pesos | r | 4905 | 4845 | 4980 |
| | dm | 1302.19 | 1362.7 | 1263.85 |
| | s | 1495.55 | 1533.16 | 1467.82 |
| | cv | 0.1977 | 0.2040 | 0.1955 |
| Forma de la distribución de los pesos | $curtosis$ | 171.02 | 161.66 | 184.94 |
| Muestra del espacio de soluciones | vo | 8023320 | 5840849 | 4096121 |
| Comportamiento del algoritmo | p_f | 100 | 100 | 100 |
| | p_uf | 0 | 0 | 0 |
| | ro | 0 | 0 | 0 |

| | | | | |
|--------------------------------------|----------------------------------------|----------------------|----------------------|-----------------------|
| | <i>delta</i> | 1 | 1 | 1 |
| Trayectoria del algoritmo | <i>nc</i> | 0.98 | 0.98 | 0.98 |
| | <i>nv</i> | 0.06 | 0.04 | 0.04 |
| | <i>pp</i> | 0.5 | 0.54 | 0.48 |
| | <i>pn</i> | 0.48 | 0.44 | 0.50 |
| Desempeño algorítmico multi objetivo | <i>Proporción de sol. no dominadas</i> | 0.42 | 0.64 | 0.32 |
| | <i>Hipervolumen</i> | 513231298226.517 | 513278774235.003 | 513599570205.834 |
| | <i>Spread</i> | 39.7311811933996 | 68.3242860079803 | 27.6662580350649 |
| | <i>Spacing</i> | 0.000104680462973524 | 0.000254480650303477 | 0.0000772010656403887 |

Una vez finalizado el proceso de caracterización, se inicia con la identificación de las mejores variantes para la solución del conjunto de algoritmos a través de criterios impuestos por el investigador.

4.6.2 Resultados obtenidos de la Identificación de las Regiones de Dominancia

En este punto, se identifican las variantes (regiones) de dominancia del algoritmo NSGA-II, a través del desempeño obtenido de cada una de ellas. En la Tabla 30 se muestran los métodos que conforman cada una de las variantes del algoritmo NSGA-II.

Tabla 30 Variantes del Algoritmo NSGA-II.

| Variante | Solución Inicial | | Método de Cruza | | Método de muta | |
|----------|------------------|------------|-----------------|-----|----------------|-----|
| | Aleatoria | Heurística | MC1 | MC2 | MM1 | MM2 |
| V1 | X | | X | | X | |
| V2 | X | | X | | | X |
| V3 | X | | | X | X | |
| V4 | X | | | X | | X |
| V5 | | X | X | | X | |
| V6 | | X | X | | | X |
| V7 | | X | | X | X | |
| V8 | | X | | X | | X |

Donde:

- MC1= Cruza OrderCrossOver
- MC2= Cruza de dos puntos
- MM1= Muta Inserción
- MM2= Muta de dos flips

Como se describe en la Tabla 30, cada variante está conformada por una combinación de los métodos del algoritmo solucionador, las cuales representan regiones que serán sometidas a evaluación para la identificación de las variantes superiores. El Algoritmo 15 muestra el proceso de identificación de las variantes del algoritmo NSGA-II propuesto en este trabajo de tesis, dando como resultado los valores mostrados en la Tabla 31 y 32.

Algoritmo 15 Proceso de Identificación de Regiones Superiores del Algoritmo NSGA-II.

1. **Inicio**
2. **Ciclo** i a $numInstancias$

3. **Ciclo** j a $numVariantes$
4. **Obtención** del desempeño de las soluciones no dominadas
5. **Guardar** el desempeño de las soluciones $\rightarrow FPV_j$
6. **Ciclo** k a 30
7. **Aplicación** del método `fastNonDominated()`
8. **Obtención** del desempeño de las soluciones no dominadas
9. **Guardar** el desempeño de las soluciones de la variante $j \rightarrow FPOV_j$
10. **Fin** del ciclo k
11. **Llamar** `EvaluacionCriterio(FPV_j, FPOV_j)`
12. **Si** la solución en FPV_j existe en $FPOV_j$
13. `contador++`
14. `matrizVariante[i][j]=contador`
15. **Si** `matrizVariante[i][j]>3`
16. `matrizBinaria[i][j]=1`
17. **Fin** ciclo j
18. **Fin** ciclo i
19. **Fin**

Tabla 31 Cantidad de Soluciones No Dominadas en el Frente de Pareto Optimo por Variante.

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----|----|----|----|----|----|----|----|
| o9p100_0 | 10 | 7 | 3 | 1 | 1 | 7 | 2 | 3 |
| o9p100_1 | 0 | 5 | 1 | 1 | 0 | 2 | 0 | 0 |
| o9p100_2 | 0 | 1 | 0 | 0 | 0 | 9 | 2 | 0 |

Como se observa en la Tabla 31, cada variante posee una cantidad de elementos que pertenecen al Frente de Pareto Optimo (FPO), se compara la cantidad obtenida y se valida si cumple con los criterios necesarios, como ejemplo didáctico se toma la variante 2, las instancias o9p100 0 y o9p100 1 tiene 7 y 5 elementos pertenecientes en el FPO, por lo tanto se asigna a cada uno el valor de 1 indicando que cumplieron el criterio, pero para la instancia o9p100 2 muestra que solamente hay 1 solución que pertenece al FPO, por lo tanto se asigna un 0 debido a que no cumple el criterio.

Tabla 32 Regiones de Dominancia de las Variantes del Algoritmo NSGA-II.

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----|----|----|----|----|----|----|----|
| o9p100_0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| o9p100_1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| o9p100_2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Una vez identificadas las variantes binarias, se seleccionan donde hay más instancias resueltas, tomando como ejemplo la Tabla 32, se seleccionaron las variantes 2, 6, 7 debido a que ellas predominan en dos o más instancias. Después de la identificación de las regiones, se inicia el proceso de selección de índices significativos.

4.6.3 Resultados obtenidos de la Selección de Índices Significativos

En esta sección se muestra el proceso de identificación de índices significativos a través de un proceso general y/o visual aplicado a índices de la estructura del problema, indica del espacio de soluciones, comportamiento y trayectoria del algoritmo (ver Algoritmo 7). En la Figura 46 y 47, se muestran de manera gráfica índices de la estructura del problema, utilizados en el proceso de caracterización del algoritmo NSGA-II, tomando como base la variante 6.

Tomando como ejemplo el índice t de la Figura 46, se muestra en promedio el espacio que ocupa un objeto dentro de un contenedor, indicando para la instancia o9p100_2 el mejor resultado calculado para el índice t demostrando que en dicha instancia hay objetos con menor peso en comparación a las demás instancias.

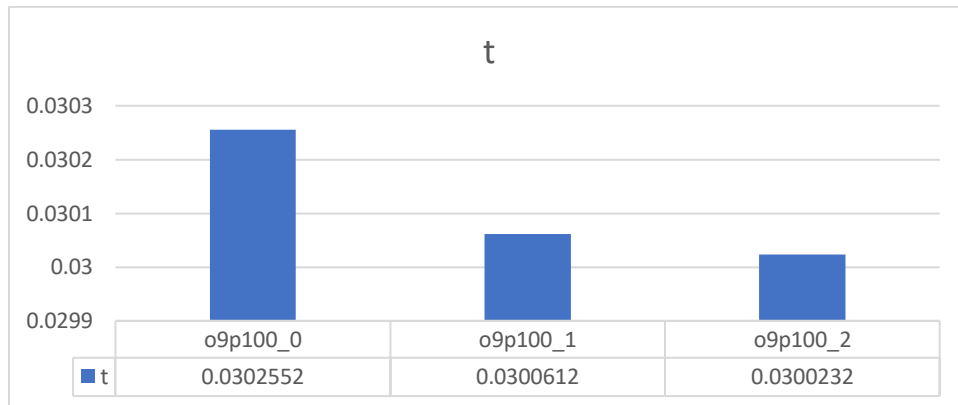


Figura 46 Gráfica de la Capacidad Ocupada del Objeto de la Variante 6.

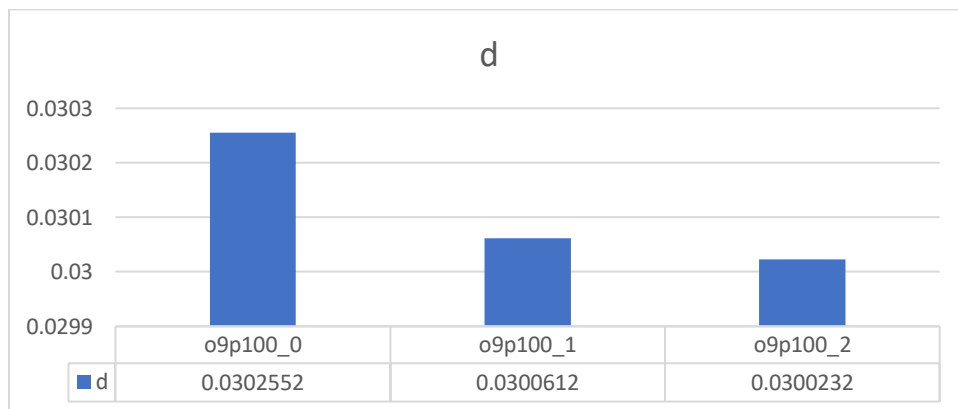


Figura 47 Gráfica de la Dispersión de los Pesos de la variante 6.

En la Figura 48 se representa de manera gráfica el índice de la muestra del espacio de soluciones del problema, tomando como base la variante 6, el cual consiste en la caracterización del espacio de soluciones. Para este caso se tomó como base el costo de cada proyecto, mostrando como resultado en la instancia o9p100_0 un mayor espacio de soluciones.

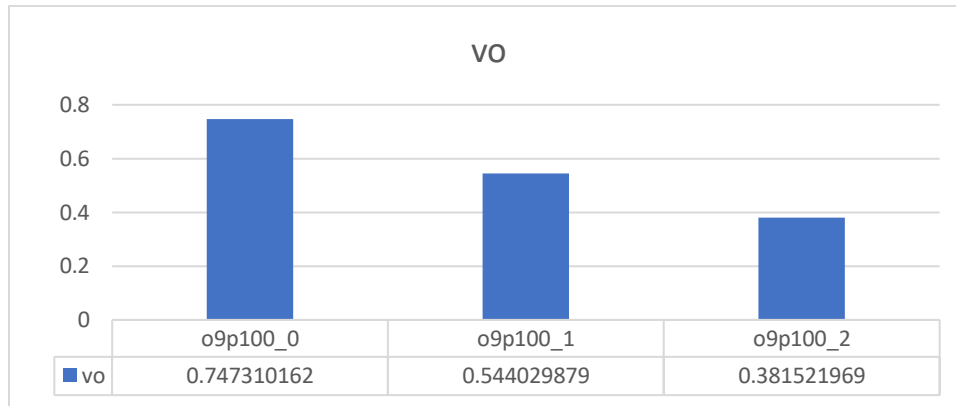


Figura 48 Gráfica de la Muestra del Espacio de Soluciones de la Variante 6.

En la Figura 49 se muestra de manera gráfica el índice *pp* de la trayectoria del algoritmo NSGA-II, tomando como base la variante 6. El índice consiste en el cálculo de pendientes positivas, dando a entender que en la instancia o9p100_1 de la variante 6, hay más cambios positivos con respecto a la disminución del costo.

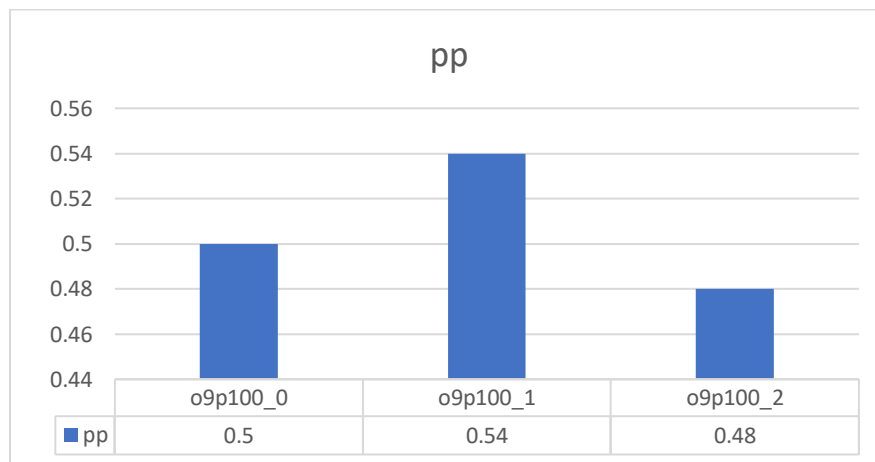


Figura 49 Gráfica del Índice de la Trayectoria del Algoritmo de la Variante 6.

Una vez identificado el conjunto de índices significativos, se inicia con la creación del modelo causal tomando como información de entrada los índices seleccionados y las regiones de dominancia obtenidas.

4.6.4 Resultados obtenidos del Análisis Causal

En este punto se muestra el modelo causal formado con los índices significativos obtenidos de la anterior fase, a través del algoritmo PC. En la Figura 50 se visualiza el modelo causal obtenido para el rediseño causal del algoritmo NSGA-II.

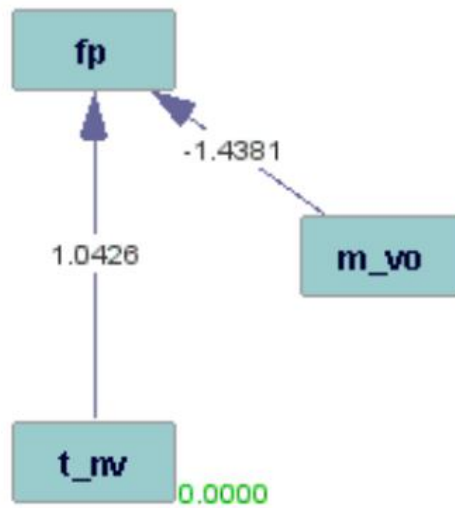


Figura 50 Modelo Causal obtenido para el Algoritmo NSGA-II.

La Figura 50 muestra que el desempeño del algoritmo (fp) es afectado por los índices número de valles (t_nv) y la muestra del espacio de soluciones del algoritmo (m_vo). En la Tabla 33 se describen las ecuaciones obtenidas del modelo causal.

Tabla 33 Ecuaciones del Modelo Causal del Algoritmo NSGA-II.

| |
|--------------------|
| fp= -1.4381 (m_vo) |
| fp=1.0426 (t_nv) |

Una vez obtenido el modelo causal, se procede a rediseñar el algoritmo siguiendo el modelo sugerido del algoritmo PC (Figura 50).

4.6.5 Resultados obtenidos del Rediseño Causal

En esta sección se muestran los resultados obtenidos de la aplicación del rediseño causal de los algoritmos metaheurístico poblacional evolutivo, aplicado al problema de la mochila mono objetivo y NSGA-II aplicado al problema de selección de cartera de proyectos multi objetivo.

El proceso del rediseño causal fue aplicado al conjunto de variantes del algoritmo solucionador (ver Tabla 16, Tabla 30), para mejorar su desempeño, siguiendo el modelo causal sugerido por Tetrad (ver Figura 34 y Figura 50).

4.6.5.1 Resultados obtenidos del Rediseño del Algoritmo Metaheurístico Poblacional Evolutivo

A continuación se describen los resultados obtenidos del proceso del rediseño del algoritmo metaheurístico poblacional, enfocado en la mejora del beneficio obtenido por variante.

En la Tabla 34 se muestra el desempeño obtenido del conjunto de variantes mostradas en la Tabla 16, del algoritmo metaheurístico poblacional sin la aplicación del rediseño causal, mostrando también los mejores resultados, reportados en la literatura para cada instancia.

Tabla 34 Resultados obtenidos del Desempeño Algorítmico sin Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional.

| Óptimos | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1453 | 1441 | 1433 | 1438 | 1439 | 1435 | 1441 | 1437 | 1444 |
| 13314829 | 13126783 | 13242591 | 13175880 | 13283926 | 13069580 | 13042620 | 12942411 | 13084730 |
| 4922 | 4910 | 4922 | 4511 | 4538 | 3954 | 4265 | 4081 | 3989 |
| 14621 | 14436 | 14626 | 8739 | 8148 | 7049 | 7012 | 7382 | 7146 |
| Óptimos | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 |
| 1453 | 1430 | 1439 | 1444 | 1435 | 1443 | 1438 | 1437 | 1443 |
| 13314829 | 13286236 | 13152358 | 13315819 | 13184053 | 12947886 | 13031783 | 12997068 | 12886286 |
| 4922 | 4914 | 4915 | 4546 | 4428 | 4128 | 4000 | 3901 | 3831 |
| 14621 | 14481 | 14516 | 8068 | 8413 | 7213 | 9306 | 7091 | 7125 |

La tabla 35 muestra el porcentaje de error de los resultados obtenidos de la Tabla 34, comparando el valor calculado de cada variante con el mejor valor reportado en la literatura.

Tabla 35 Resultados obtenidos del Porcentaje de Error sin Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|-------|-------|-------|--------|--------|--------|--------|--------|
| 1.166 | 1.715 | 1.372 | 1.303 | 1.578 | 1.166 | 1.44 | 0.96 |
| 3.117 | 2.262 | 2.755 | 1.957 | 3.539 | 3.738 | 4.478 | 3.427 |
| 0.244 | 0 | 8.35 | 7.802 | 19.667 | 13.348 | 17.087 | 18.956 |
| 1.696 | 0.402 | 40.49 | 44.515 | 51.999 | 52.251 | 49.731 | 51.338 |
| V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 |
| 1.92 | 1.303 | 0.96 | 1.578 | 1.029 | 1.372 | 1.44 | 1.029 |
| 1.94 | 2.928 | 1.722 | 2.694 | 4.437 | 3.818 | 4.074 | 4.892 |
| 0.163 | 0.142 | 7.639 | 10.037 | 16.132 | 18.732 | 20.744 | 22.166 |
| 1.92 | 1.303 | 0.96 | 1.578 | 1.029 | 1.372 | 1.44 | 1.029 |

LA Tabla 36 indica el desempeño obtenido de cada variante del algoritmo metaheurístico poblacional después de la aplicación del rediseño causal, mostrando también los mejores resultados, reportados en la literatura para cada instancia.

Tabla 36 Resultados obtenidos del Desempeño Algorítmico con Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional.

| Óptimos | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1453 | 1453 | 1447 | 1453 | 1450 | 1444 | 1443 | 1446 | 1449 |
| 13314829 | 13207294 | 13274499 | 13283926 | 12942411 | 13126783 | 13122122 | 13287358 | 13265162 |
| 4922 | 4879 | 4911 | 4499 | 4562 | 4730 | 4827 | 3979 | 4028 |
| 14621 | 14292 | 14303 | 8687 | 9226 | 8443 | 9306 | 7537 | 7520 |
| Óptimos | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 |
| 1453 | 1443 | 1442 | 1453 | 1453 | 1451 | 1444 | 1453 | 1400 |
| 13314829 | 13271572 | 13292134 | 13314765 | 13302829 | 13271108 | 13141174 | 13235578 | 13279414 |
| 4922 | 4872 | 4884 | 4759 | 4702 | 4464 | 4612 | 4030 | 4027 |

| | | | | | | | | |
|-------|-------|-------|------|------|------|------|------|------|
| 14621 | 14320 | 14355 | 9253 | 9124 | 9446 | 8881 | 7515 | 7642 |
|-------|-------|-------|------|------|------|------|------|------|

En la Tabla 37 se muestra el porcentaje de error obtenido de cada variante del algoritmo solucionador después de la aplicación del rediseño causal.

Tabla 37 Resultados obtenidos del Porcentaje de Error con Rediseño Causal del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|-------|--------|--------|--------|--------|--------|--------|--------|
| 0 | 0.413 | 0 | 0.206 | 0.619 | 0.688 | 0.482 | 0.275 |
| 0.808 | 0.303 | 0.233 | 2.877 | 1.433 | 1.447 | 0.206 | 0.373 |
| 0.874 | 0.223 | 8.594 | 7.314 | 3.901 | 1.93 | 19.159 | 18.163 |
| 2.25 | 2.175 | 40.585 | 36.899 | 42.254 | 36.352 | 48.451 | 48.567 |
| V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 |
| 0.688 | 0.7571 | 0 | 0 | 0.1376 | 0.6194 | 0 | 3.6476 |
| 0.325 | 0.1704 | 0.0005 | 0.0901 | 0.3284 | 1.3042 | 0.5952 | 0.266 |
| 1.016 | 0.772 | 3.3117 | 4.4697 | 9.3052 | 6.2983 | 18.123 | 18.184 |
| 2.059 | 1.8193 | 36.714 | 37.597 | 35.394 | 39.259 | 48.601 | 47.733 |

Como se ve en la Tabla 35 y 37, el porcentaje de error varía dependiendo de la instancia que se analizó, debido al número de elementos que contiene cada instancia o a los métodos que conforman la variante del algoritmo solucionador. Para el caso de la Tabla 35 y 37 las variantes 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15 y 16 muestran un porcentaje de error mayor en comparación a los valores calculados del resto del conjunto de variantes, demostrando que su desempeño no fue muy bueno.

Al tener soluciones con un porcentaje de error muy pequeño, puede dificultar el rediseño, debido a que no hay mucho margen para que el proceso de rediseño permita mejorar su desempeño.

La Tabla 38 muestra el porcentaje de mejora del desempeño del conjunto de variantes del algoritmo solucionador, comparando los resultados mostrados en la Tabla 34 y 36.

Tabla 38 Resultados obtenidos del Porcentaje de Mejora del Desempeño del Conjunto de Variantes del Algoritmo Metaheurístico Poblacional.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.83276 | 0.97697 | 1.04312 | 0.76442 | 0.62718 | 0.13879 | 0.62630 | 0.34626 |
| 0.61333 | 0.24095 | 0.82003 | -2.57089 | 0.43768 | 0.60956 | 2.66525 | 1.37895 |
| -0.63136 | -0.22349 | -0.26602 | 0.52887 | 19.62570 | 13.17702 | -2.49939 | 0.97769 |
| -0.99751 | -2.20840 | -0.59503 | 13.23024 | 19.77585 | 32.71535 | 2.09970 | 5.23370 |
| V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 |
| 0.90909 | 0.20848 | 0.62327 | 1.25436 | 0.55440 | 0.41725 | 1.11343 | -2.97990 |
| -0.11037 | 1.06274 | -0.00792 | 0.90091 | 2.49633 | 0.83942 | 1.83511 | 3.05075 |
| -0.85470 | -0.63072 | 4.68544 | 6.18790 | 8.13953 | 15.30000 | 3.30684 | 5.11616 |
| -1.11180 | -1.10912 | 14.68765 | 8.45121 | 30.95799 | -4.56695 | 5.97941 | 7.25614 |

En la Tabla 38 el proceso de rediseño causal permitió mejorar el desempeño de ciertas variantes tomando como ejemplo, las variantes 5, 6, 8, 11, 12, 13, 15, 16 ya que fueron variantes en donde el rediseño causal mejoro el desempeño en todas las instancias. Además se demostró que el proceo de

rediseño permitió la mejora del desempeño de variantes, las cuales se encuentran resaltadas con borde negro.

En ciertos casos el rediseño no mejoro el desempeño de todas las instancias o la mejora no fue significativa, esto es debido a que era muy cercano al optimo reportado de la literatura, es decir que no había un margen grande de mejora (ver Tabla 35 y Tabla 37).

4.6.5.2 Resultados obtenidos del Rediseño del Algoritmo NSGA-II

A continuación se presentan los resultados obtenidos del proceso del rediseño del algoritmo NSGA-II, enfocado en mejorar el número de soluciones encontradas en el frente de Pareto y la cantidad de soluciones que pertenecen al frente de Pareto óptimo.

En la Tabla 39 se muestran los resultados obtenidos de la aplicación del índice de la proporción de soluciones no dominadas (RNI) aplicado al conjunto de variantes del algoritmo solucionador sin el rediseño causal.

Tabla 39 Resultados obtenidos de la Proporción de Soluciones No Dominadas sin la aplicación del Rediseño Causal del Conjunto de Variantes del Algoritmo NSGA-II.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|------|------|------|------|------|------|------|------|
| 0.37 | 0.47 | 0.78 | 0.43 | 0.6 | 0.22 | 0.2 | 0.11 |
| 0.58 | 0.57 | 0.44 | 0.45 | 0.48 | 0.54 | 0.6 | 0.61 |
| 0.55 | 0.46 | 0.72 | 0.36 | 0.38 | 0.49 | 0.55 | 0.56 |

La Tabla 40 muestra los resultados obtenidos del índice RNI, aplicado al conjunto de variantes del algoritmo solucionador con el rediseño causal.

Tabla 40 Resultados obtenidos de la Proporción de Soluciones No Dominadas después de la aplicación del Rediseño Causal del Conjunto de Variantes del Algoritmo NSGA-II.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|------|------|------|------|------|------|------|------|
| 0.6 | 0.43 | 0.65 | 0.44 | 0.6 | 0.33 | 0.35 | 0.24 |
| 0.69 | 0.49 | 0.48 | 0.35 | 0.5 | 0.43 | 0.61 | 0.34 |
| 0.4 | 0.37 | 0.45 | 0.38 | 0.55 | 0.51 | 0.52 | 0.34 |

Por último en la Tabla 41 se muestra el porcentaje de mejora del índice RNI, tomando como información de entrada los valores mostrados en la Tabla 37 y 38.

Tabla 41 Porcentaje de mejora del Índice de Proporción de Soluciones No Dominadas del Conjunto de Variantes del Algoritmo NSGA-II.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----------|----------|----------|----------|---------|----------|---------|----------|
| 62.1622 | -8.5106 | -16.6667 | 2.3256 | 0.0000 | 50.0000 | 75.0000 | 118.1818 |
| 18.9655 | -14.0351 | 9.0909 | -22.2222 | 4.1667 | -20.3704 | 1.6667 | -44.2623 |
| -27.2727 | -19.5652 | -37.5000 | 5.5556 | 44.7368 | 4.0816 | -5.4545 | -39.2857 |

Como se muestra en la Tabla 41 el proceso de rediseño permitió mejorar en algunos casos el porcentaje de soluciones del frente de Pareto, solamente la variante 5 fue la única en donde el proceso de rediseño causal mejoró la proporción de soluciones en todas las instancias.

En las variantes 1,6,7 el rediseño mejoró la proporción de soluciones no dominadas en al menos dos instancias, mostrando en algunas un buen porcentaje de mejora, y en otras un porcentaje bajo de mejora.

Y por último las variantes 2, 3 muestran que el proceso de rediseño no fue efectivo debido a que empeoró el porcentaje de soluciones en el frente cero.

En la Tabla 42 se presentan los resultados obtenidos del cálculo de número de soluciones por variante que pertenecen al frente de Pareto óptimo, sin la aplicación del rediseño causal.

Tabla 42 Resultados obtenidos de la Cantidad de Soluciones pertenecientes del Frente de Pareto Optimo sin la aplicación del Rediseño Causal.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----|----|----|----|----|----|----|----|
| 12 | 7 | 2 | 7 | 28 | 27 | 15 | 13 |
| 47 | 30 | 8 | 9 | 29 | 40 | 28 | 21 |
| 37 | 24 | 30 | 13 | 35 | 19 | 40 | 15 |

En la Tabla 43 se muestra los resultados obtenidos del cálculo de número de soluciones que pertenecen al frente de Pareto optimo, aplicado al conjunto de variantes del algoritmo solucionador después de la ejecución del rediseño causal.

Tabla 43 Resultados obtenidos de la Cantidad de Soluciones pertenecientes del Frente de Pareto Optimo después de la aplicación del Rediseño Causal.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|----|----|----|----|----|----|----|----|
| 41 | 17 | 29 | 22 | 34 | 47 | 24 | 20 |
| 43 | 23 | 25 | 14 | 49 | 29 | 24 | 26 |
| 29 | 27 | 20 | 10 | 41 | 38 | 19 | 25 |

Por último en la Tabla 44 se muestra el porcentaje de mejora del número de soluciones que pertenecen al frente de Pareto optimo, tomando como información de entrada los valores mostrados en la Tabla 40 y 41.

Tabla 44 Porcentaje de mejora del Número de Soluciones pertenecientes al Frente de Pareto Optimo del Conjunto de Variantes del Algoritmo NSGA-II.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 |
|-----------|-----------|------------|-----------|----------|-----------|-----------|----------|
| 241.66667 | 142.85714 | 1350.00000 | 214.28571 | 21.42857 | 74.07407 | 60.00000 | 53.84615 |
| -8.51064 | -23.33333 | 212.50000 | 55.55556 | 68.96552 | -27.50000 | -14.28571 | 23.80952 |
| -21.62162 | 12.50000 | -33.33333 | -23.07692 | 17.14286 | 100.00000 | -52.50000 | 66.66667 |

Por último en la Tabla 44, el proceso del rediseño causal permitió mejorar en ciertos casos, el número de soluciones por variante que pertenecen al frente de Pareto optimo; por ejemplo, en las variantes 5 y 8 el porcentaje de mejora en todas las instancias fue positivo, indicando que el rediseño mejoró la calidad de la solución.

En el caso de las variantes 1, 2, 3, 4, 6 en ciertas instancias el porcentaje de mejora fue mayor o igual al 100%, esto quiere decir que la aplicación del rediseño causal permitió mejorar de manera exponencial el número de soluciones que pertenecen al frente de Pareto óptimo.

Capítulo 5: Conclusiones y Trabajos Futuros

En este capítulo se presenta a manera de conclusión, las aportaciones realizadas en este trabajo de investigación y la posible continuidad a futuro para los siguientes investigadores.

5.1 Conclusiones

Las conclusiones a las que se llegaron en este trabajo de investigación fueron las siguientes:

Se cumplió con el objetivo principal del mejoramiento del desempeño de algoritmos mono y multi objetivo dando solución al problema de la mochila mono objetivo y el problema de selección de cartera de proyectos multi objetivo, a través de una metodología causal la cual se incorporó a la herramienta VisTHAA. Dicho objetivo no se pudo cumplir sin el rediseño del módulo de preprocesamiento de instancias para la incorporación de problemas multi objetivo y de más problemas mono objetivo y la incorporación de más índices de la estructura del problema, del comportamiento, trayectoria y desempeño algorítmico.

El rediseño fue mostrado de manera visual a través de una comparación entre el desempeño obtenido en las variantes que no fueron seleccionadas en la fase de identificación de regiones de dominancia y los resultados obtenidos en el rediseño.

Las principales contribuciones son las siguientes:

1. Diseño de una metodología general para el preprocesamiento de instancias multi objetivo.
2. Rediseño del módulo de preprocesamiento para la aceptación de problemas multi objetivo y más problemas mono objetivo.
3. Rediseño del módulo de caracterización de la estructura del problema para instancias multi objetivo y de más problemas mono objetivo las cuales no se encontraban incorporadas en la herramienta.
4. Incorporación de más índices de la estructura del problema en VisTHAA.
5. Incorporación de índices del comportamiento, trayectoria y desempeño algorítmico en VisTHAA.
6. Incorporación de una metodología causal para la herramienta de VisTHAA.
7. Diseño e implementación de un módulo de caracterización del comportamiento, trayectoria y desempeño algorítmico.
8. Diseño e implementación del módulo de rediseño causal para algoritmos mono/multi objetivo.

5.2 Trabajos Futuros

En esta sección se sugieren los siguientes trabajos futuros y líneas de investigación que complementarían la herramienta VisTHAA, las cuales se mencionan a continuación:

1. Implementación de un módulo de incorporación de nuevos problemas a estudiar.
2. Incorporación de más índices del proceso de caracterización para la herramienta VisTHAA.
3. Incorporación de más pruebas estadísticas para la validación de resultados.
4. Incorporación de más algoritmos para la solución de problemas mono/ multi objetivo.
5. Incorporación de nuevas funciones que permitirán robustecer la herramienta VisTHAA.
6. Rediseño del módulo de caracterización de instancias para los nuevos problemas a analizar.
7. Rediseño del módulo de rediseño causal para los nuevos problemas y algoritmos a estudiar.

Referencias

| | |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Alvarez06] | Álvarez V. Modelo para representar la Complejidad del problema y el desempeño de algoritmos. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2006. |
| [Baldi12] | Mauro Maria Baldi · Teodor Gabriel Crainic · Guido Perboli · Roberto Tadei (2012) “Branch-and-price and beam search algorithms for the Variable Cost and Size Bin Packing Problem with optional items” |
| [Bartz10] | Thomas Bartz-Beielstein, Marco Chiarandini, Luis Paquete.: “ <i>Experimental Methods for the Analysis of Optimization Algorithms</i> ”. Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-02537-2 and e-ISBN 978-3-642-02538-9. (2010). |
| [Basse98] | Basse S. Computer Algorithms, Introduction to Design and Analysis. Editorial Addison-Wesley Publishing Company, 1998. |
| [Bastiani15] | Bastiani, S. S., Cruz-Reyes, L., Fernandez, E., & Gomez, C., <i>Portfolio Optimization From a Set of Preference Ordered Projects Using an Ant Colony Based Multi-objective Approach. International Journal of Computational Intelligence Systems</i> , 8 (sup2), 41-53, 2015. |
| [Berenson01] | Berenson Mark L., Levine, David M. Estadística para administración, Segunda Edición. Pearson Educación, México, 2001 |
| [Carazo10] | Carazo, A. F., Gómez, T., Molina, J., Hernández-Díaz, A. G., Guerrero, F. M., & Caballero, R., <i>Solving a Comprehensive Model for Multiobjective Project Portfolio Selection. Computers & Operations Research</i> , 630–639, 2010. |
| [Carnegie06] | Carnegie Mellon University. Causal and Statistical Reasoning. Open Learning Initiative, http://oli.web.cmu.edu/openlearning/ |
| [Castillo11] | Norberto C Carnegie06castillo Garcia “Evaluación de Estrategias de Mejora del Desempeño de Metaheurísticos Aplicados a BPP Vía Diagnóstico Visual” Instituto Tecnológico de Cd. Madero 2011 |
| [Cerecero15] | Cerecedo, C. Jorge A., <i>Algoritmo de Procesamiento Celular para Solución del Problema de Cartera de Proyectos</i> , Instituto Tecnológico de Ciudad Madero, 2015. |
| [Chevalier06] | Chevalier R. Balancing the effects of parameter settings on a genetic algorithm for multiple fault diagnosis. Artificial Intelligence, The university of Georgia, 2006. |
| [Chickering95] | Chickering, D.: A transformational Characterization of Equivalent Bayesian Network Structures. 11 the Conference on Uncertainly AI. San Francisco, pp. 87-98 |
| [Chuin05] | Hoong Chuin Lau, Wee Chong Wan, Steven Halim.: “Tuning Tabu Search Strategies via Visual Diagnosis”. School of Information Systems. Singapore Management University. (2005). |
| [Cohen95] | Cohen P. R. “Empirical Methods for Artificial Intelligence”. The MIT Press, Cambridge, Massachusetts, 1995. |
| [Coello07] | Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen “Evolutionary Algorithms for Solving Multi-Objective Problems” Springer 2007 |
| [Crainic11] | Crainic, T. G., Perboli, G., Rei, W., & Tadei, R. (2011). <i>Efficient lower bounds and heuristics for the variable cost and size bin packing problem. Computers and Operations Research</i> , 38(11), 1474–1482. |
| [Cruz04] | Cruz L. Caracterización de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas. Tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México, 2004. |
| [Deb00] | K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, <i>Proceedings of the Parallel Problem Solving from Nature VI Conference</i> , pages 849–858, Paris, France, 2000. Springer |
| [Deb01] | Deb, K. <i>Multi-Objective Optimization Using Evolutionary Algorithms</i> . 2001. |

| | |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Deb02] | Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. <i>Evolutionary Computation, IEEE Transactions on</i> , 6(2), 182-197 |
| [Dijkstra61] | Dijkstra E. W.: "ALGOL-60 Translation". Stichting Mathematisch Centrum. 2e Boerhaavestraat 49. Amsterdam. Rakenafdeling. (1961). |
| [Dogan08] | U., Dogan A.: "An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms". <i>Advances in Engineering Software</i> , 40 (5). (2008). |
| [Duarte06] | Duarte, A., Pantrigo, J., and Gallego, M. <i>Metaheurísticas</i> . Universidad Rey Juan Carlos, 2006. |
| [Duarte07] | Duarte Muñoz, A., & Pantrigo Fernandez, Juan Jose Gallego Carrillo, M. (2007). <i>Metaheurísticas</i> . <i>Dykinson</i> , 53(9), 1689–1699. |
| [Esposito97] | Esposito, F., aleaba, D., Ripa V., Semeraro G.: <i>Discovering Causal Rules in Relational Databases</i> . <i>Applied Artificial Intelligence</i> vol. 11 , pp. 71-84 (1997). |
| [Fernandez01] | Eduardo Fernández González, Jorge Navarro Castillo Universidad Autónoma de Sinaloa 2001 |
| [Fernandez03] | Fernandez, E., and Navarro, J. A genetic search for exploiting a fuzzy preference model of portfolio problem with public projects. Kluwer Academic Publishers (2003). |
| [Fernandez09] | E. Fernández, L. F. Felix and G. Mazcorro, "Multi-objective Optimisation of an Outranking Model for Public Resources Allocation on Competing Projects", <i>International Journal of Operational Research</i> , Vol. 5, No. 2, pp. 190-210, 2009. |
| [Fuente05] | Antonio la Torre de la Fuente "Algoritmos Geneticos Paralelos" 2005 |
| [Garey79] | Michael R. Garey "Computers and Intractability a Guide to the Theory of NP-Completeness" 1979 <i>Variable neighborhood search for the variable sized bin packing problem</i> . <i>Computers and Operations Research</i> , 39(5), 1097–1108 |
| [Ghasemzadeh00] | F. Ghasemzadeh, N.P. Archer "Project portfolio selection through decision support" Michael G. DeGroote School of Business, McMaster University, Hamilton, ON, Canada L8S 4M4, February 2000. |
| [Glymour99] | Glymour C., Cooper, G. <i>Computation, Causation & Discovery</i> . AAAI Press/ The MIT Press. 1999 |
| [Gomez10] | Gómez, C. G., Cruz, L., Meza, E., Schaeffer, E., Castilla, G.: "A Self-Adaptative Ant Colony System for Semantic Query Routing Problem in P2P Networks". In: <i>Computación y Sistemas</i> Vol. 13 No. 4, pp 433-448. ISSN: 1405-5546. (2010). |
| [Gomez14] | María del Carmen Gómez Fuentes, Jorge Cervantes Ojeda (2014) <i>Introducción al análisis y al diseño de algoritmos</i> |
| [Hemmelmayr11] | Vera Hemmelmayr, V. C. (2011). Variable neighborhood search for the variable sized bin packing problem. <i>ELSEVIER</i> , 12 |
| [Hoos04] | Hoos H., Smyth K., Stützle T. Search Space Features Underlying the Performance of Stochastic Local Search Algorithms for MAX-SAT. <i>Parallel Problem Solving from Nature VIII, Lecture Notes in Computer Science</i> , 3242: 51-60, 2004. |
| [Holland95] | Holland, J.H.: <i>Adaptation in Natural and Artificial Systems</i> . University of Michigan Press, Ann Arbor (1975) |
| [Jiang14] | Jiang, S., Ong, Y. S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. <i>IEEE Transactions on Cybernetics</i> , 44(12), 2391-2404 |
| [Kang03] | Kang, J., & Park, S. (2003). Algorithms for the variable sized bin packing problem. <i>European Journal of Operational Research</i> , 147(2), 365–372 |

| | |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Konak05] | Konak A., Kulturel S. Simulation optimization using Tabu search: An empirical study. Proceedings of Winter simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2005. |
| [Landro08] | Landro V. Desarrollo de un Método Formal que Muestre la Interrelación entre las Características de un Conjunto de Casos y las de un Algoritmo que los Resuelve Eficientemente, para el Problema de Distribución de Objetos en Contenedores. Tesis de doctorado, Dpto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2008. |
| [Lawler85] | E.L. Lawler, J.K. Lenstra, A.H.G.Rinnooy Kan, D.B. Shmoys The Traveling Salesman Problem: G.E. Re Guided Tour of Combinatorial Optimization Wiley and Sons, New York (1985) |
| [Lemeire04] | Lemeire J., Dirx E. <i>Causal Models for Parallel Performance Analysis</i> . 4th PA3CT Symposium, Edegem, Belgium, 2004. |
| [Lemeire05] | Lemeire J., Dirx E. <i>Causal Models for Parallel Performance Analysis</i> . 4th PA3CT Symposium, Edegem, Belgium, 2005. |
| [Liu07] | D.S. Liu, K.C. Tan, S.Y. Huang, C.K. Goh, W.K. Ho Department of Electrical and Computer Engineering, National University of Singapore, 4, Engineering Drive 3, Singapore 117576, Singapore 2007 |
| [Maarten04] | Maarten Vankerkom Jin Yu.: “Visualizing Swarm algorithms”. Tesis de Maestría. Faculteit Toegepaste Wetenschappen. (2004). |
| [Madsen05] | Madsen A., Jensen F., Kjaerulff U., Lang M. <i>The Hugin Tool for Probabilistic Graphical Models</i> . International Journal on Artificial Intelligence Tools, 14(3): 507-544, 2005. |
| [Martello90] | Martello, S., and Toth, P. <i>Knapsack problems: algorithms and computer implementations</i> . John Wiley & Sons, 1990. |
| [Martínez10] | Martínez Blanca “Análisis causal de un algoritmo híbrido metaheurística basado en Búsqueda Tabú” Instituto Tecnológico de Cd. Madero 2010 |
| [Martinez17] | Enith Martinez Cruz “Adaptación de un algoritmo de agrupación mono-objetivo a multi objetivo usando caracterización del desempeño” |
| [McGeoch00] | McGeoch, C.C.: “Experimental Analysis of Algorithms”. In: Pardalos, P.M., Romeijn, H.E.: <i>Handbook of Global Optimization</i> , Vol. 2, pp. 489-513. (2000). |
| [Mendenhall97] | William Mendenhall, Terry Sincich.: “ <i>Probabilidad y Estadística para Ingeniería y Ciencias</i> ”. Fourth Edition. Prentice-Hall Hispanoamérica, S.A. ISBN (translation): 968-880-960-8. ISBN (original): 0-02-380581-1. (1997). |
| [Mladenovic97] | Mladenovic, N., & Hansen, P. (1997). Variable Neighborhood Search, 24(1), 1097–1100. |
| [Mustafa17] | Mustafa Avci, Seyda Topaloglu , A Multi-Start Iterated Local Search Algorithm for the Generalized Quadratic Multiple Knapsack Problem, <i>Computers and Operations Research</i> (2017) |
| [Nebro04] | Nebro, A., Alba, E., and Luna, F. Optimización multi-objetivo y computación grid. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Malaga, 2004. |
| [Norsys09] | Norsys Software Corp. Netica Application. 2009. |
| [Padilla18] | Alejandro Estrada Padilla “Optimización del problema de empaqueo de contenedores con costos y tamaños variable” Tesis de Maestría del Instituto Tecnológico de Cd. Madero 2018 |

| | |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Perez07] | Pérez, V. Modelo Causal de Desempeño de Algoritmos Metaheurísticos en Problemas de Distribución de Objetos. PhD thesis, Instituto Tecnológico de Ciudad Madero, 2007. Ciudad Madero, Tamaulipas, México. |
| [Perez11] | Pérez j. “Towards a Characterization of Difficult Instances of the Bin Packing Problem” 2011 |
| [Polya45] | Polya, G. (1945). How to Solve It. <i>The Mathematical Gazette</i> |
| [Quilan93] | Quinlan, J. R. 1993. C4.5: Programs for machine learning. San Mateo, Calif.: Morgan Kaufmann. |
| [Quiroz09] | Marcela Quiroz Castellanos “Caracterización de Factores de Desempeño de Algoritmos de Solución De BPP” Instituto Tecnológico de Cd. Madero 2009 |
| [Ramírez10] | Ramírez Gabriela “Análisis causal de un algoritmo híbrido metaheurístico basado en genéticos” Instituto Tecnológico de Cd. Madero 2010 |
| [Schrijver98] | Schrijver, A. <i>Theory of Linear and Integer Programming</i> . John Wiley & Sons, 1998 |
| [Seshadri] | Aravind Seshadri Multi-Objective Optimization Using Evolutionary Algorithms (Moea) |
| [Sprites01] | Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, MIT Press, 2nd edition (2001). |
| [Srinivas94] | N. Srinivas and Kalyanmoy Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, <i>Evolutionary Computation</i> 2 (1994), no. 3, 221 – 248. |
| [Tan02] | K.C. TAN1, T.H. LEE and E.F. KHOR “ <i>Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons</i> ” <i>Department of Electrical and Computer Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260</i> . |
| [Ugur08] | Ugur A.: “ <i>Path Planning On A Cuboid Using Genetic Algorithms</i> ”. <i>Information Sciences</i> 178, pp. 3275-3287. (2008). |
| [Ugur10b] | Ugur A., TSPAntSim, software disponible en línea, http://yzgrafik.ege.edu.tr/projects/TSPAntSim/ , último acceso: 19/Septiembre/2011. |
| [Ugur11a] | Ugur A., TSPAntSim, software disponible en línea, http://yzgrafik.ege.edu.tr/projects/TSPAntSim/ , último acceso: 19/Septiembre/2011. |
| [Ugur11b] | Ugur A., CuboidTSP Versión 2.0, software disponible en línea, http://yzgrafik.ege.edu.tr/~ugur/CuboidTSP/ , último acceso: 19/Septiembre/2011. |
| [Van98] | Van Veldhuizen, D. A., & Lamont, G. B. (1998). <i>Multiobjective evolutionary algorithm research: A history and analysis</i> . Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. |
| [Wang05] | Wang, L., Zhang, Y., & Feng, J. (2005). On the Euclidean distance of images. <i>IEEE transactions on pattern analysis and machine intelligence</i> , 27(8), 1334-1339. |
| [Weinberg90] | Weinberg, E. D. Correlated and uncorrelated fitness landscapes and how to tell the difference. <i>Biological Cybernetics</i> , 63, 325- 336. 1990 |
| [Weiss07] | Weiss, M. A. <i>Data Structures and Algorithm Analysis in Java</i> , 2nd ed. Pearson-Addison Wesley, 2007 |
| [Zitzler99] | Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. <i>Evolutionary computation, IEEE transactions on</i> , 3(4), 257-271. |
| [Zitzler00] | Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. <i>Evolutionary computation</i> , 8(2), 173-195. |

