INSTITUTO TECNOLÓGICO
DE CD. MADERO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Modelado Causal del Desempeño de Algoritmos Metaheurísticos en Problemas de Distribución de Objetos

Presenta:
ISC. Verónica Pérez Rosas

Para obtener el grado de:
Maestra en Ciencias en Ciencias de la Computación

Director:
Dra. Laura Cruz Reyes

Codirector:
MC. Claudia Guadalupe Gómez Santillán

CD. MADERO, TAM. MÉXICO. DICIEMBRE DE 2007

D.I.

Instituto Tecnológico de Cd. Madero

Cd. Madero, Tam., a 03 de Diciembre de 2007.

Área: Posgrado
N° Oficio: U5.471/07
Asunto: Autorización de Impresión de Tesis

C. ING. VERÓNICA PÉREZ ROSAS
P r e s e n t e.

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

"MODELADO CAUSAL DEL DESEMPEÑO DE ALGORITMOS METAHEURÍSTICOS EN PROBLEMAS DE DISTRIBUCIÓN DE OBJETOS"

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.

A t e n t a m e n t e
"POR MI PATRIA Y POR MI BIEN"

M.P. María Yolanda Chávez Cinco
Jefa de la División

ESTADOS UNIDOS MEXICANOS
S.E.P.
DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION
I.T.C.M.

MYCHC ' NLCO ' cerc*

DEDICATORIA

Dedico este trabajo a dos personas muy especiales… mi padre y Oscar.

Gracias papá porque has sido el mejor de los ejemplos

Gracias Oscar porque iniciamos este proyecto juntos y fuiste mi motivación y mi apoyo para conseguir esta meta tan importante en mi vida

AGRADECIMIENTOS

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) y a la Dirección General de Educación Tecnológica (DGEST) por el apoyo y facilidades recibidas para la conclusión de este trabajo.

Mi más sincero aprecio y agradecimiento a la Dra. Laura Cruz Reyes, porque además de ser mi guía académica, fue un ejemplo de perseverancia y esfuerzo constante. No olvidare las lecciones aprendidas ni sus valiosos consejos.

Agradezco también al Instituto Tecnológico de Cd. Madero por las facilidades proporcionadas para el desarrollo de este proyecto, así como a la planta de profesores que contribuyeron a mi formación académica y humana.

Gracias a los miembros del comité de tesis: Dr. Arturo Hernández Ramírez, Dra. Laura Cruz Reyes, MC. Claudia G. Gómez Santillán y Dr. José A. Martínez, por su colaboración e interés en este trabajo. Así como también al Dr. Hector Fraire Huacuja, Dr. Joaquín Pérez Ortega y MC. Vanesa Landero Nájera por sus valiosas críticas y comentarios.

Un reconocimiento especial a la Dra. Elisa Satu Schaeffer, profesor investigador del Programa de Posgrado en Ingeniería de Sistemas de la Universidad Autónoma de Nuevo León, por colaborar en la revisión de esta tesis.

Gracias a todo el equipo que participó en el desarrollo de este trabajo, en especial a Víctor M. Alvarez Hernández y a Gilberto Rivera Zarate por su dedicación y esfuerzo.

Gracias a toda mi familia, por su paciencia y comprensión, ustedes son mi mayor motivo para salir adelante y seguir superándome..

Gracias a mis compañeros de la maestría por su compañerismo y apoyo. En especial gracias a Tania y Bárbara, por sus consejos, sincera amistad y apoyo constante.

RESUMEN

Debido a la complejidad de muchos problemas del mundo real, ha sido propuesta una gran variedad de algoritmos aproximados, los cuales han mostrado un desempeño satisfactorio en la solución de problemas de optimización. Sin embargo, dado que no existe un algoritmo que sea la mejor opción para todas las posibles situaciones, es necesario elegir el algoritmo más adecuado para un problema específico. Un obstáculo que se presenta es identificar cuál algoritmo es mejor y por qué es la mejor opción.

La construcción de modelos causales es un enfoque que ha sido propuesto, en otras áreas del conocimiento, para el estudio de fenómenos naturales y sociales. Los modelos causales son una representación formal de las relaciones causa-efecto que existen entre los elementos que integran un sistema. La aplicación estos modelos en la tarea de seleccionar el mejor algoritmo para un problema dado, permitiría aumentar la confiabilidad de la elección realizada y descubrir los factores que causan el desempeño.

Este trabajo presenta una metodología para el análisis experimental de un tipo especial de algoritmos aproximados, llamados metaheurísticos. El propósito del análisis es identificar aspectos que influyen en su desempeño y establecer relaciones entre ellos para explicar su comportamiento al resolver un problema real. El problema abordado es la distribución de objetos en contenedores, que consiste en asignar un conjunto de objetos en la mínima cantidad de contenedores.

La metodología plantea un estudio sistemático de los elementos que intervienen en el proceso algorítmico asociado a una metaheurística. El estudio involucra tres etapas: diseño experimental, creación de modelos causales y análisis estadístico de éstos para proporcionar explicaciones del comportamiento observado. Se aplicó la metodología al análisis del desempeño de un algoritmo de Búsquela Tabú. Las principales relaciones encontradas revelan que este algoritmo se desempeña mejor cuando se incorporan estrategias que le permiten una mayor exploración del espacio de búsqueda, como iniciar con una solución totalmente aleatoria, y utilizar varios operadores de búsqueda local.

SUMMARY

Due to real World problems complexity, a variety of approximated algorithms which had shown satisfactory performance in optimization problems had been proposed. However, there is not an algorithm that performs better for all possible situations, given the amount of available strategies, is necessary to select the one who adapts better to the problem. An important point is to know which strategy is the best for the problem and why is it better.

An approach that recently has been proposed to answer this question is the generalization of prediction models through causal models. Which are formal representations for cause effect relations that exist between observed elements in a system. Appling this kind of models to select the best algorithm for a given problem allows increasing selection reliability.

A methodology to experimental analysis for a special sort of approximated algorithms called metaheuristics is presented in this work. The purpose is to identify factors that influence its performance and to establish relations between them to explain its behavior in a real world problem. Investigation problem is minimizing the number of containers needed to distribute an objects set.

A systematic study of elements that take place in algorithmic process is proposed in the methodology. The study consists in three phases: experimental design, causal models construction and statistical analysis of them to provide explanations about observed behavior. This methodology was applied to analyze Tabu Search algorithm performance. The main discovered causal relations reveal that Tabu Search performs better when the incorporated search strategies permit more space search exploration as initializing search with a completely random solution and to apply several operators in local search.

CONTENIDO

INDÍCE DE FIGURAS

INDÍCE DE TABLAS

Capítulo 1

INTRODUCCIÓN

La investigación sobre la metodología de experimentación computacional está creciendo rápidamente, y tiene como objetivo promover que los experimentos sean relevantes, correctos, replicables y que produzcan conocimiento [McGeoch00].

Trabajos recientes en el área de las ciencias computacionales han demostrado que no es suficiente saber que un algoritmo es superior a otro en la solución de un conjunto particular de instancias de un problema. También es de interés contar con explicaciones del comportamiento observado.

Los problemas de aplicación real son muy variados y las técnicas aplicadas para solucionarlos deben de ser, preferiblemente, adaptadas a la naturaleza del problema. Trabajos anteriores analizaron el desempeño de los algoritmos bajo el concepto de caja negra, en el que los aspectos a analizar fueron, principalmente, la adaptabilidad al problema y la competitividad de los resultados obtenidos. Sin embargo, se han presentado nuevas necesidades en el uso y aplicación de algoritmos complejos a problemas del mundo real. Por ejemplo, conocer la naturaleza e interpretación de las relaciones presentes entre los elementos que intervienen en el desempeño.

Esta situación hace necesario analizar no solo el problema a resolver, sino también saber qué sucede en el interior del algoritmo. El objetivo principal es encontrar relaciones entre el problema y el comportamiento observado y también conocer el impacto de estas relaciones en el desempeño obtenido.

Este trabajo trata el estudio y análisis experimental del desempeño de algoritmos metaheurísticos que resuelven problemas complejos, como lo es el problema de empacado de objetos: Bin Packing. Se propone para estos fines, aplicar un enfoque sistemático de identificación de relaciones causa-efecto entre problema, algoritmo y desempeño. Este enfoque es conocido en la literatura especializada como modelado causal y es aplicado ampliamente en áreas como ciencias sociales y biológicas.

1.1 ANTECEDENTES

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. A través del análisis y tratamiento de la información obtenida, es posible encontrar patrones que extraigan algún tipo de conocimiento útil para las organizaciones.

Generalmente, en el área de minería de datos el método más utilizado para el hallazgo de patrones son los algoritmos de clasificación y predicción: árboles de reglas, clasificadores bayesianos, redes neuronales y reglas de asociación entre otros. Sin embargo, en su mayoría los modelos generados por este tipo de algoritmos son predictivos u asociativos y frecuentemente dependientes de los datos a partir de los cuales fueron generados. Esto implica que no tienen la capacidad para explicar de manera confiable el patrón observado y no es posible realizar una generalización del conocimiento obtenido.

Teniendo como objetivo la explicación y generalización del comportamiento observado, se ha incursionado en el desarrollo y aplicación de algoritmos que proporcionen explicaciones confiables del comportamiento observado. Mediante las explicaciones es posible obtener mayor conocimiento de los sistemas y por lo tanto brindar retroalimentación acerca de cómo mejorar su funcionamiento.

Este trabajo presenta la aplicación de un enfoque causal para explicar el desempeño de algoritmos metaheurísticos. El objetivo de este trabajo es contribuir a un mayor conocimiento de los algoritmos identificando los factores que causan y afectan su desempeño.

1.2 DESCRIPCIÓN FORMAL DEL PROBLEMA DE INVESTIGACIÓN

Dado un conjunto de grupos $G = \{g_1, g_2, ..., g_n\}$ de casos de un problema de optimización P y un conjunto de algoritmos $A = \{a_1, a_2, ..., a_n\}$, donde cada $a \in A$ resuelve mejor los casos de un grupo $g \in G$.

Se pretende caracterizar a los conjuntos P y A para obtener indicadores de la complejidad del problema *IP*, del comportamiento del algoritmo *IC*, y del desempeño del algoritmo *ID*. Para asociar un modelo causal $M = (IP, IC, ID, F)$ que represente relaciones funcionales F entre los indicadores *IP*, *IC* e I*D*.

La caracterización del problema y del algoritmo es una parte esencial en el análisis del desempeño de los algoritmos, y permite identificar cuáles son las características (indicadores) que los describen adecuadamente. El modelo causal *M* provee una representación formal de las relaciones existentes entre los indicadores de la complejidad del problema, y el comportamiento y desempeño del algoritmo. La interpretación del modelo podría responder al cuestionamiento de por qué un algoritmo se comporta mejor con un determinado conjunto de instancias de un problema dado.

La Figura 1.1 muestra como se relacionan los elementos que intervienen en el proceso de construcción del modelo causal *M*. El recuadro Problema muestra grupos $\{g_1, g_2\} \in G$ de casos que son resueltos por los algoritmos Búsqueda Tabú y Aceptación por Umbral. En el recuadro Algoritmo se muestra, para cada algoritmo, parte de las trayectorias generadas al solucionar las instancias $i_1 \in g_1$ y $j_1 \in g_2$. En el recuadro desempeño se presentan las soluciones asociadas a los puntos finales de las trayectorias mencionadas.

En la secuencia mostrada en la Figura 1.1, se puede notar que el desempeño del algoritmo (calidad de las soluciones encontradas), depende en cierto grado de la trayectoria que sigue el algoritmo, y ésta a su vez, se relaciona con la naturaleza del problema que resuelve.

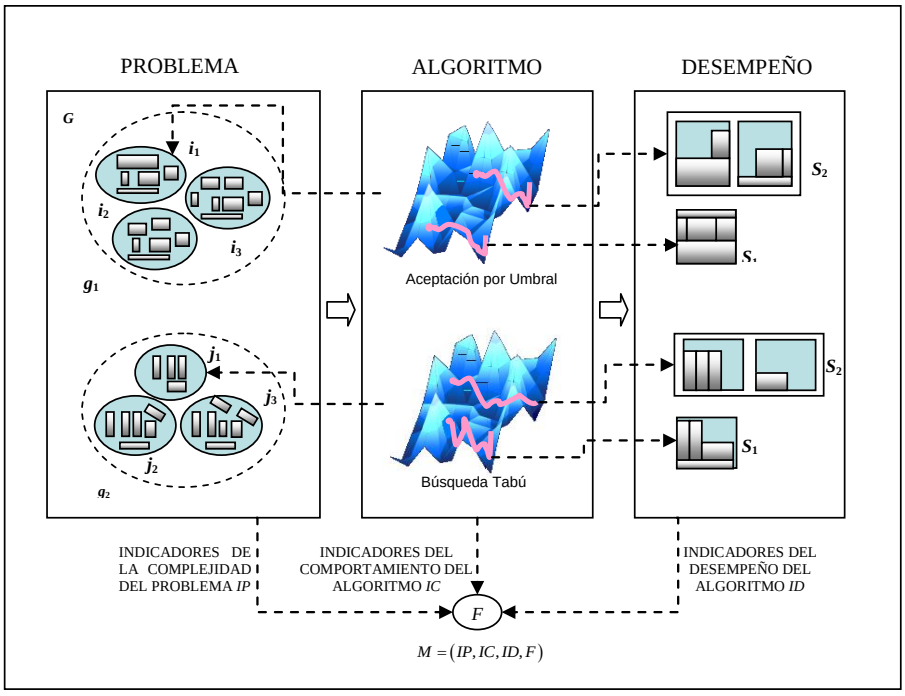


Figura 1.1 Problema de Investigación

1.3 HIPÓTESIS

Dado un conjunto de factores que caracterizan el problema, el comportamiento de un algoritmo sobre ese problema y el desempeño obtenido por el mismo. ¿Es posible encontrar, mediante un método formal, relaciones entre ellos que expliquen por qué un algoritmo es más adecuado para resolver un conjunto particular de casos?

1.4 JUSTIFICACIÓN

El principal objetivo del análisis de la información, es el descubrimiento de conocimiento útil. Éste debe ser transportable y en el mejor de los casos, independiente del contexto, permitiendo así su generalización y aplicación a diversas problemáticas.

Si se obtiene conocimiento acerca de las relaciones causales entre las características de los datos es posible utilizarlo en crear guías o políticas para tratar una situación [Glymour&Cooper99]. Por ejemplo, se puede analizar una base de datos con la intención de aprender algo acerca de los factores que influyen en la demanda de energía eléctrica de un grupo de consumidores.

Trabajos recientes en las ciencias computacionales han incursionado en la aplicación de modelos causales para el análisis de algoritmos de solución para problemas complejos. Se considera que contar con un modelo de esta naturaleza aportaría elementos para un mayor entendimiento del comportamiento de los algoritmos utilizados. El conocimiento obtenido podría ser utilizado para generalizar los modelos de predicción del desempeño, lograr un incremento en la exactitud de la predicción e incluso el rediseño de algoritmos y sus indicadores de desempeño.

El proceso de crear un modelo causal no es trivial éste, según la literatura, es un problema NP completo [Chickering95] lo cual implica una alta complejidad computacional. La construcción de modelos causales consiste principalmente en la

búsqueda a través del espacio de todos los posibles modelos que se ajusten a los datos y esto implica que para un conjunto de datos lo suficientemente grande el costo computacional, es exponencial.

1.5 OBJETIVOS

A continuación se presentan el objetivo general y los objetivos específicos planteados para esta investigación.

1.5.1 OBJETIVO GENERAL

Analizar el desempeño de algoritmos metaheurísticos en problemas de distribución de objetos utilizando un enfoque causal

1.5.2 OBJETIVOS ESPECÍFICOS

- Disponer de un conjunto de algoritmos metaheurísticos diseñados, implementados y evaluados en las mismas condiciones.
- Identificar y seleccionar factores relacionados con el desempeño de un algoritmo metaheurístico.
- Encontrar relaciones entre los factores que influyen en el desempeño de un algoritmo metaheurístico.
- Diseñar una metodología de construcción de modelos causales para la explicación del desempeño algorítmico.
- Desarrollar un modelo causal que explique el desempeño de algoritmos metaheurísticos en la solución del problema clásico de empacado de objetos en contenedores (Bin Packing).
- Interpretar el modelo obtenido y generar explicaciones del por qué un algoritmo se comporta mejor sobre un conjunto particular de instancias.

1.6 ORGANIZACIÓN DEL DOCUMENTO

El documento esta organizado de la siguiente manera: el Capítulo 2 presenta una revisión de los fundamentos teóricos para el análisis experimental de algoritmos metaheurísticos. Se describe el problema de empacado de objetos y las estrategias metaheurísticas aplicadas para resolver este problema. Se presentan también los fundamentos teóricos del análisis de la superficie de aptitudes, que fue el enfoque utilizado para describir el comportamiento de los algoritmos. Finalmente se describe la evolución de las estrategias de análisis experimental que han sido propuestas para explicar el comportamiento de los algoritmos.

El Capítulo 3 presenta una visión general de la estrategia de modelado causal, describiendo para esto los conceptos teóricos necesarios para su aplicación. Se presenta también una reseña de los trabajos relacionados con la aplicación de estos modelos en la minería de datos, así como los enfoques propuestos para su creación. Finalmente se muestra un análisis comparativo de las estrategias de modelado causal que han sido propuestas para explicar el comportamiento de los algoritmos.

En el Capítulo 4 se presenta de manera detallada la metodología propuesta, cuyas principales etapas son: diseño experimental, creación de modelos causales y el análisis de éstos para proporcionar explicaciones del comportamiento observado.

En el Capítulo 5 se presentan los resultados experimentales de la aplicación de la metodología propuesta para el análisis del desempeño del algoritmo Búsqueda Tabú. Obteniendo como resultado un conjunto de explicaciones causales que relacionan los elementos que afectan el desempeño del algoritmo y explican su comportamiento.

En el Capítulo 6 se presentan las conclusiones de la investigación realizada, así como sugerencias para trabajos futuros.

Capítulo 2

ANÁLISIS EXPERIMENTAL DEL DESEMPEÑO DE ALGORITMOS METAHEURÍSTICOS

En este capítulo se presenta una revisión de los conceptos y enfoques relacionados con el análisis experimental del desempeño de algoritmos metaheurísticos que resuelven problemas complejos.

2.1 INTRODUCCIÓN

La mayoría de los trabajos relacionados con el análisis del desempeño de algoritmos metaheurísticos, están enfocados en su demostración a partir de resultados experimentales comparativos. Sin embargo, trabajos recientes en el área han demostrado la necesidad y utilidad de explicar el desempeño observado. Esta tarea consiste de analizar y comprender a los factores que afectan o intervienen en el desempeño obtenido por un algoritmo.

El análisis individual de estos factores puede mostrar ciertas influencias para aquellos casos, en los que, por ejemplo, el desempeño fue adecuado o inadecuado. El inconveniente de este análisis es que no muestra con exactitud las interacciones entre

factores. Conocerlas es necesario para proporcionar explicaciones, ya que éstas permiten conocer la naturaleza de la relación encontrada.

Independientemente del método elegido para la caracterización de los algoritmos, el objetivo principal es entender como el desempeño de un algoritmo depende de una serie de factores que lo influencian. El entendimiento adquirido puede conducir a mejores predicciones del desempeño en nuevas situaciones y al descubrimiento de algoritmos mejorados.

2.2 PROBLEMA DE EMPACADO DE OBJETOS EN CONTENEDORES: BIN PACKING

La definición de este problema fue tomada de [Cruz04]. El problema de distribución de objetos en contenedores, en ingles Bin Packing, es un problema clásico de optimización combinatoria NP-duro. En este trabajo se resuelve la versión discreta de este problema en su variante de una dimensión. La expresión 2.1 presenta la definición formal del problema de Bin Packing.

Dados (2.1)

n = número de objetos a distribuir

c = capacidad del contenedor

L = secuencia de n objetos a_i

$s_i(a_i)$ = tamaño de cada objeto a_i

Encontrar una partición de L mínima, $l = B_1 \cup B_2 \cup,...,\cup B_m$ tal que en cada conjunto B_j la sumatoria del tamaño de cada objeto $s(a_i)$ en B_j no exceda c.

Tal que:

$$\sum_{a_i \varepsilon B_j} s_i(a_i) \leq c \qquad \forall j,\ 1 \leq j \leq m.$$

2.3 ALGORITMOS METAHEURÍSTICOS PARA LA SOLUCIÓN DEL PROBLEMA BIN PACKING

Debido a que Bin Packing es un problema de alta complejidad no es posible resolver casos grandes utilizando un algoritmo exacto. Según [Basse98] una solución óptima puede encontrarse considerando todas las formas de hacer una partición del conjunto de n objetos en un subconjunto de tamaño n o más pequeño; desafortunadamente el número de posibles particiones es mayor que $\left(n/2\right)^{n/2}$.

De acuerdo con [Papadimitriou98] cuando se tiene un problema clasificado como NP duro y se desea resolver un caso grande del mismo, la única opción posible es "dar una buena solución", pero no necesariamente la mejor. Los algoritmos metaheurísticos, han mostrado un buen desempeño para encontrar soluciones aproximadas aceptables para problemas complejos [Pérez04].

2.3.1 DEFINICIÓN DE ALGORITMO METAHEURÍSTICO

Un algoritmo metaheurístico es un algoritmo de propósito general, que tiene incorporadas ciertas heurísticas definidas de acuerdo al problema y cuyo comportamiento es no determinista; es decir no encuentra la misma solución al ejecutarse varias veces con la misma entrada. Dadas sus características, este tipo de algoritmos han sido muy estudiados por la comunidad científica en su aplicación a problemas complejos del mundo real.

Los algoritmos metaheurísticos más reconocidos y utilizados son, entre otros, Búsqueda Tabú, Aceptación por Umbral, Recocido Simulado, y Optimización por Colonia de Hormigas [Glover86, Perez04, Falkenauer06]. Este tipo de algoritmos, consisten básicamente en un proceso iterativo, dirigido por los siguientes elementos [Fonlup97]:

a. **Estado inicial**. También llamado solución inicial, es un punto en el espacio de búsqueda del problema a partir del cual inicia el proceso. Puede ser obtenido de manera aleatoria o utilizando una estrategia heurística.
b. **Estado Actual**. Es un punto o colección de puntos en el espacio de búsqueda del problema.
c. **Vecindad**. Es el conjunto de puntos o soluciones definidas a partir de un operador de búsqueda que es aplicado al estado inicial para obtener un conjunto de estados siguientes (soluciones). Un operador de búsqueda puede estar compuesto por varios operadores e incluso puede ser una metaheurística.
d. **Regla de Transición**. Es la estrategia que permite seleccionar el siguiente estado a partir de la vecindad generada.
e. **Criterio de Parada**. Es el criterio que permite que el algoritmo se detenga. Comúnmente se fija tomando en cuenta la convergencia del algoritmo a una solución aceptable o a un número de iteraciones establecidas.

2.3.2 ALGORITMO BÚSQUEDA TABÚ (TABU SEARCH)

En este algoritmo, el proceso inicia con una solución factible $x^* \in X$ donde X es el conjunto de todas las soluciones factibles del problema, y continua buscando iterativamente un mínimo global, transitando entre soluciones vecinas que no sean tabú (prohibidas). Para esto: N es una función que asocia a cada elemento en X una vecindad predefinida (por ejemplo, el conjunto de todas las soluciones factibles que se pueden obtener intercambiando pares de objetos), y *LTabu* es una lista que "memoriza" temporalmente los movimientos tabú.

El proceso detallado de la búsqueda iterativa es el siguiente: se genera una lista de soluciones candidatas *LCandi*, tomadas de la vecindad $N(x^*)$; se selecciona la mejor solución que no sea tabú y se convierte temporalmente en movimiento prohibido agregándolo a *Ltabu*. A continuación se actualiza el tiempo de permanencia t de todas

las soluciones en la lista tabú, quitando la prohibición a las que cumplieron su tiempo de estancia; finalmente, si $z(x) < z(x^*)$ x^* se actualiza como mejor solución a x.

2.4 DESEMPEÑO EN ALGORITMOS METAHEURÍSTICOS

El desempeño de un algoritmo basado en heurísticas esta determinado por su eficiencia y su efectividad. La efectividad de un algoritmo se refiere a la calidad de la solución encontrada o a su confiabilidad en la tarea de encontrar soluciones adecuadas. La eficiencia por otra parte, esta caracterizada por el comportamiento del algoritmo en tiempo de ejecución, por ejemplo, el tiempo computacional o requerimientos de memoria.

Sin embargo, este análisis no es siempre aplicable, principalmente porque los algoritmos heurísticos son considerados muchas veces como cajas negras cuyo funcionamiento interno no es conocido. Otro aspecto a considerar es que la complejidad del problema resuelto dificulta la estimación de la eficiencia.

La efectividad de un algoritmo es altamente dependiente de la estructura del problema y esto tiene que ver directamente con las instancias del problema que se analiza. La eficiencia, por otra parte, esta muy relacionada con el conocimiento que se tenga del dominio del problema. Para el análisis del desempeño de los algoritmos metaheurísticos, la comunidad científica muestra mayor interés en estudiar los aspectos relacionados con su efectividad [Merz98]. Este aspecto es el más complicado de modelar debido a que interviene de manera directa la naturaleza del problema.

McGeoch [McGeoch00] y Barr [Barr95] sugieren la existencia de tres categorías principales de factores que afectan el desempeño algorítmico: problema, algoritmo y ambiente.

a. *Factores del problema*: dimensión, distribución de los parámetros que lo definen, estructura del espacio de solución entre otros.

b. *Factores del algoritmo*: estrategias heurísticas seleccionadas (procesos de construcción de solución inicial y parámetros de búsqueda asociados), códigos de computadoras empleados, configuración de control interno del algoritmo y comportamiento en la ejecución entre otros.
c. *Factores del ambiente*. Estos factores se refieren al ambiente físico en el que serán ejecutados los algoritmos como los son el software (sistema operativo, compilador) y hardware (velocidad del procesador, memoria). Así como también aquellos relacionados con el programador: pericia, habilidad de afinación, lógica.

Los criterios para medir el desempeño de los algoritmos de aproximación dependen de los métodos elegidos para su caracterización, que pueden ser teóricos ó experimentales. En los primeros, para cada algoritmo, se determina matemáticamente la cantidad de recursos necesarios como función del tamaño del caso considerado mejor, peor o promedio. Los segundos se basan en la experimentación para realizar la caracterización y a diferencia de los anteriores permiten describir el comportamiento de casos específicos.

En los algoritmos metaheurísticos raramente se estudia su efectividad de manera teórica, esto debido principalmente a la complejidad de los problemas de combinatoria. Asumiendo que no hay algoritmos que resuelvan en tiempo polinomial los problemas denominados NP duros, la única opción disponible es analizar experimentalmente la efectividad de los algoritmos metaheurísticos.

2.5 ANÁLISIS DE LA SUPERFICIE DE APTITUDES DE ALGORITMOS METAHEURÍSTICOS

La dificultad asociada al espacio de búsqueda de un problema se relaciona con la trayectoria que describen las soluciones de un algoritmo. Es posible que ésta muestre una gran cantidad de óptimos locales, por lo que la dificultad de encontrar buenas

soluciones a través de un proceso de optimización dependería de la calidad de las soluciones y de la distribución y accesibilidad de cada óptimo local [Smith02].

Una pregunta fundamental para este análisis es: ¿cuál es la influencia de la complejidad del problema en la trayectoria descrita por el algoritmo a través del espacio de solución del problema? En la literatura se encuentran propuestas para este cuestionamiento, principalmente trabajos enfocados a lo que se denomina Análisis de la Superficie de Aptitudes (Fitness Landscape).

El análisis de la superficie de aptitudes es un enfoque utilizado inicialmente en computación evolutiva. Fue propuesto por Wright [Wright32] y se basa en considerar a la evolución como el flujo de una población sobre una superficie. Donde la altitud de un punto cuantifica que tan bien se adapta al ambiente un individuo.

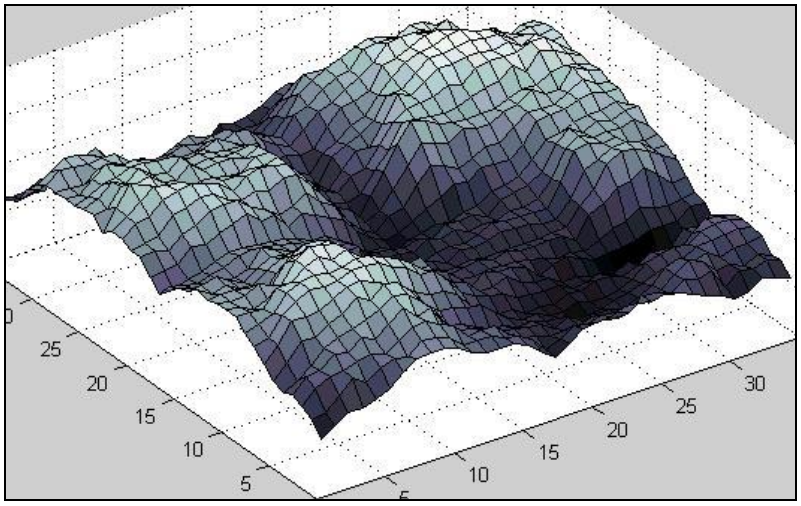


Figura 2.1 Superficie de aptitudes del espacio de búsqueda

La superficie de aptitud es considerada como la visualización del proceso de exploración y explotación del espacio de búsqueda, definido a través de los individuos (soluciones), imaginándolo como una superficie formada por picos, valles, montañas,

ríos, y planicies, ver Figura 2.1. Un valor de aptitud es asociado a cada individuo y una función de aptitud califica en cierto grado si es apto o no. El flujo de la población esta definido por un operador de evolución, de modo que las conexiones entre los individuos generados están determinadas por el operador empleado para la búsqueda a través de la superficie.

2.5.1 CONCEPTOS RELACIONADOS

Para el análisis de una superficie de aptitud asociada a un algoritmo, se consideran tres elementos: la representación del conjunto de soluciones, la función de aptitud que evalúa a las soluciones y los operadores que definen las relaciones de vecindad entre el conjunto de soluciones [Vassilev99].

La función de aptitud toma la representación de la solución y la trasforma en un valor de aptitud, evalúa la solución para el problema dado y retorna un número que indica qué tan buena es la solución. El proceso evolutivo, visto de forma general, es un proceso de maximización, por lo que usualmente se asigna mayor valor de aptitud a las mejores soluciones [Hordijk95].

Para un problema de optimización como Bin Packing cuyo objetivo es la minimización, la función de aptitud toma como entrada el número de contenedores utilizados y la distribución de los objetos. Una forma de asociar el valor de aptitud es evaluando la calidad de la distribución de los objetos. De manera que mientras mejor sea el acomodo de los mismos, menor la cantidad de contenedores requeridos y mayor el valor de aptitud asignado.

La estructura de una superficie de aptitudes esta especificada en función de la rugosidad o neutralidad que describe. El término rugosidad se refiere a las diferencias que existen entre valores de aptitud de una vecindad de individuos y la neutralidad

describe las áreas que se consideran planas, es decir aquellas en las que no se aprecian diferencias en la vecindad generada.

A continuación se proporciona una breve definición de los términos utilizados para describir una superficie de aptitudes [Kallel01]:

- Óptimo global. Es el punto con mejor valor de aptitud en la superficie de aptitudes.
- Óptimo local. Es un punto es la superficie de aptitudes cuyo valor de aptitud es mejor que el de todos los puntos cercanos a el.
- Pico. Es un óptimo local de una función de maximización
- Valle. Es un óptimo local en una función de minimización.
- Valle de atracción de un óptimo local. Es un conjunto m_j de puntos $x_1, x_2, ..., x_k$ del espacio de búsqueda, tal que un algoritmo de búsqueda local iniciando a partir de x_i con $1 \leq i \leq k$ termina en el óptimo local m_j.
- Modalidad. Es el número y tamaño de los valles de atracción.

2.5.2 MÉTRICAS ESTADÍSTICAS

Función de Auto-correlación (autocorrelation Function ACF)

Weinberg [Weinberg90] propuso esta métrica para medir la rugosidad de una superficie de aptitudes. Se considera que existe un fuerte enlace de este concepto y la dureza de un problema de optimización para un algoritmo basado en búsqueda local. Es decir, que intuitivamente el número de óptimos locales depende del enlace (correlación) entre el costo de una solución y el costo de sus vecinos.

Para su cálculo, dadas las aptitudes $f_1, f_2, f_3, ..., f_n$ de un conjunto de individuos, donde n es el número de aptitudes medidas a una distancia *k*. La función de auto-correlación ρ_k se define en las expresiones 2.2 y 2.3.

$$\overline{f} = \frac{1}{n}\sum_{i=1}^{n} f, n > 0 \tag{2.2}$$

$$\rho_k = \frac{\sum_{i=1}^{N-k}\left(f_i - \overline{f}\right)\left(f_{i+k} - \overline{f}\right)}{\sum_{i=1}^{N}\left(f_i - \overline{f}\right)^2}, n > k \tag{2.3}$$

Donde si $|\rho_k| \approx 1$, entonces existe mucha correlación entre los *k* puntos medidos, mientras que si $|\rho_k| \approx 0$ existe poca correlación entre los mismos. El valor de *k* más común es uno, ya que permite encontrar las correlaciones entre cada par de aptitudes consecutivas.

Un valor alto de la función de auto-correlación indica que las diferencias en la aptitud entre cualquier par de soluciones vecinas es en promedio muy similar, implicando una superficie menos rugosa. En caso contrario, un valor cercano a 0 indica que los valores de aptitud son casi independientes, y por tanto la superficies es muy rugosa y tentativamente mas difícil.

Longitud de auto-correlación (Autocorrelation lenght AC)

Esta métrica, también propuesta por Weinberg [Weinberg90] indica cuál es el valor mayor de distancia a la que el conjunto de soluciones se vuelve no correlacionado. La longitud de auto correlación $\|\rho_k\|$ entre un conjunto de soluciones, separadas a una distancia *k*, se calcula mediante la expresión 2.4.

$$\|\rho_k\| = \frac{1}{1-\rho_k} \tag{2.4}$$

Donde ρ_k es la función de auto-correlación entre las soluciones obtenidas con la expresión 2.3. Un valor alto de la métrica indica una superficie muy plana, mientras que un valor pequeño sugiere una superficie más rugosa.

Coeficiente de engaño (Deceptiveness)

Esta métrica se utiliza para evaluar la relación entre el problema y la trayectoria que describe un operador de búsqueda [Jones95b]. Para calcular esta métrica se aplica el concepto de *aptitud final*, que es el valor de aptitud final obtenida cuando se cumple el criterio de paro asociado a un operador de búsqueda. A partir de varias ejecuciones del algoritmo, se puede obtener el conjunto $F = \{ f_1, f_2, ..., f_m \}$ de *m* aptitudes finales.

El coeficiente de engaño se define en la expresión 2.5. Donde *E* es el valor esperado de aptitud final y los valores máximo y mínimo de *F* son $F_{mín}$ y $F_{máx}$. Si $F_{máx} - F_{mín} = 0$ entonces el valor de esta métrica es 0.

$$coef\_eng = 1 - \frac{E - F_{mín}}{F_{máx} - F_{mín}} \tag{2.5}$$

Este coeficiente se interpreta de la siguiente manera: si el valor obtenido muestra tendencia a 1 significa que el problema es engañoso, mientras que un valor cercano a 0 indica que el problema es amigable o sencillo de resolver.

2.5.3 MÉTRICAS DE INFORMACIÓN

Este enfoque se basa en la teoría clásica de la información, y tiene como objetivo cuantificar el grado de rugosidad o llanura de una superficie de aptitud. Las métricas propuestas evalúan la entropía generada en la trayectoria que siguen las soluciones encontradas utilizando un operador de búsqueda.

Basándose en este enfoque han sido propuestas las métricas: contenido de información, contenido de la información parcial e información de la densidad de valles [Vasilev99, Fogarty99]. Estas son descritas en las secciones siguientes.

***Contenido de información* (*Information Content*)**

El contenido de información caracteriza el grado de rugosidad con respecto a las áreas planas de la superficie de aptitudes. El grado de llanura detectado depende de un parámetro de sensitividad ε, que es un valor real en el rango $[0, L]$, donde L representa la máxima diferencia en la secuencia $\{f_i\}_{i=1}^{n}$.

Para calcular esta métrica, la secuencia de aptitudes $f_1, f_2, f_3, ..., f_n$ se codifica en una cadena $S(\varepsilon) \in \{\bar{1}, 0, 1\}$. Para un valor particular de ε la cadena se genera utilizando la función $s_i = \psi_{ft}(i, \varepsilon)$ descrita por la expresión 2.6.

$$\psi_{ft}(i,\varepsilon) = \begin{cases} \bar{1} & \text{si } f_i - f_{i-1} < -\varepsilon \\ 0 & \text{si } f_i - f_{i-1} \leq -\varepsilon \\ 1 & \text{si } f_i - f_{i-1} > -\varepsilon \end{cases} \tag{2.6}$$

La cadena *S* representa los posibles cambios en la trayectoria generada por las aptitudes. Estos cambios corresponden a los valles, picos, cuestas, ascensos y zonas planas que se presentan durante el recorrido. Los sub-bloques de secuencias de valores de aptitud *pq* que pueden presentarse en *S* se muestran en laTabla 2.1.

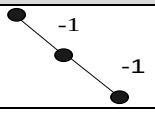
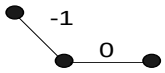
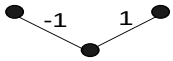
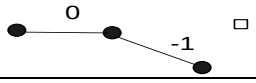
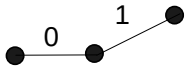
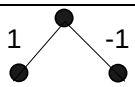
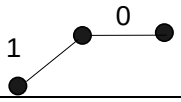
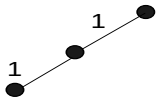
La expresión 2.7 presenta la expresión para calcular el contenido de información $H(\varepsilon)$ de la cadena $S(\varepsilon)$. Donde $P_{[pq]}$ son las probabilidades de los sub-bloques de aptitudes *pq* encontrados donde *p*≠*q*.

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \tag{2.7}$$

$H(\varepsilon)$ es una medida de la entropía del sistema, por lo que el análisis será mas sensitivo cuando el parámetro $\varepsilon = 0$, ya que produce la mayor cadena de 1's y -1's. De manera inversa, el análisis será menos sensitivo cuando $\varepsilon = L$, ya que generará una cadena de 0's al enumerar *S* y por consecuencia proveerá una descripción menos

detallada de la superficie de aptitudes. Un valor cercano a uno para esta métrica indica una superficie muy rugosa.

Tabla 2.1 Cambios en la trayectoria descritos por *S*

Sub-bloques	Forma de la trayectoria	Codificación de sub-bloques
-1-1		*pp*
-10		*pq*
-11		*pq*
0-1		*pp*
00		*pp*
01		*pq*
1-1		*pq*
10		*pq*
11		*pp*

Contenido de la información parcial (***Partial information content***)

Esta métrica es obtenida al filtrar las partes no esenciales de la cadena $S(\varepsilon)$, de manera que la sub-cadena *S'* que resulta es una indicación de la modalidad encontrada en la trayectoria y se determina mediante la expresión 2.8.

$$M(\varepsilon) = \frac{\mu}{n} \tag{2.8}$$

Donde μ es la longitud de la sub-cadena $S'(\varepsilon)$ y n es la longitud de la cadena original $S(\varepsilon)$. μ es calculada mediante la función recursiva $\phi_S(1,0,0)$ definida en la expresión 2.9.

$$\phi_S(i,j,k) = \begin{cases} k & \text{si } i > n \\ \phi_S(i+1, i, k+1) & \text{si } j = 0 \text{ y } S_i \neq 0 \\ \phi_S(i+1, i, k+1) & \text{si } j > 0, S_i \neq 0 \text{ y } S_i \neq S_j \\ \phi_S(i+1, j, k) & \text{otro caso} \end{cases} \tag{2.9}$$

Donde *k* retornará el valor de μ al completarse la evaluación. Cuando μ es 0, es indicador de que no existen pendientes en el recorrido. Cuando el valor obtenido para μ es cercano a 1, es un claro indicador de que el recorrido es en su mayor parte multimodal (muchos picos y valles).

Utilizando el valor de $M(\varepsilon)$ es posible estimar el número esperado de óptimos en la trayectoria, mediante la expresión 2.10.

$$opt\_e = \frac{nM(\varepsilon)}{2} \tag{2.10}$$

Información de la densidad de valles (***Density Basin Information***)

Esta métrica permite identificar las partes planas de la superficie de aptitudes. Da una indicación de la densidad y el aislamiento de los picos en la superficie de aptitudes. Para calcular la información de la densidad de valles se utiliza la expresión 2.11.

$$h(\varepsilon) = -\sum_{p \in \{1,0,1\}} P_{[pp]} \log_3 P_{[pp]} \tag{2.11}$$

Donde *pp* representa los posibles sub-bloques *pp*, es decir: 00, 11,-1-1 (ver Figura 2.1) en *S*. Un valor alto de $h(\in)$ es evidencia de un número elevado de picos en un área pequeña. Un número pequeño de picos existentes en un área pequeña producirían un

valor pequeño de $h(\in)$ y esto indica la presencia de óptimos aislados. De esta forma $h(\in)$ proporciona una idea de la cantidad y la naturaleza de los picos que encuentran en el camino hacia el valor óptimo.

2.6 EXPLICACIÓN DEL DESEMPEÑO DE ALGORITMOS METAHEURISTICOS

La explicación de un comportamiento observado se logra a través de la transición entre tres etapas: descripción, predicción y causalidad [Cohen95a]. La primera etapa consiste en responder a preguntas exploratorias cuya respuesta descriptiva no requiere de mucho entendimiento de la situación que genera el comportamiento observado, ya que es posible describir algo sin necesidad de comprenderlo.

La segunda etapa requiere de un mayor conocimiento ya que permite extender la respuesta a casos no conocidos, y para esto es necesario comprender las condiciones bajo las cuales el comportamiento ocurre. En la tercera etapa las respuestas son más complejas ya que se pretende dar una explicación del comportamiento observado, esto implica conocer las causas genuinas del fenómeno analizado.

Para la explicación del comportamiento de algoritmos aproximados, categoría a la cual pertenecen los algoritmos metaheurísticos, la transición entre las etapas mencionadas se ha dado gradualmente. En un principio, los trabajos reportados por la comunidad científica estaban enfocados a estudiar el desempeño demostrando la superioridad de un algoritmo a través de resultados comparativos con instancias estándar [Hooker94].

Este primer enfoque puede ser catalogado como descriptivo. Los siguientes trabajos que se presentaron contemplaban una caracterización del algoritmo y algunas observaciones respecto a su desempeño [Borghetti96].

Trabajos posteriores incorporaron la posibilidad de predicción basada en la previa caracterización del algoritmo [Cruz04]. Recientemente se ha incursionado en la creación de modelos más completos y complejos que tratan de explicar como se relacionan las características que influye en el desempeño [Lemeire04].

Capítulo 3

MODELOS CAUSALES DEL DESEMPEÑO ALGORÍTMICO

En este capítulo se presentan los fundamentos teóricos y trabajos relacionados del modelado causal del desempeño de algoritmos.

3.1 FUNDAMENTOS TEÓRICOS

3.1.1 CAUSALIDAD

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 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.

3.1.2 MODELOS CAUSALES

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 si mismo, y 3) si *A* es causa de *B* entonces *B* no es causa de *A*.

La representación de un sistema causal se hace 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.

3.1.3 DEFINICIONES PARA EL MODELADO CAUSAL

Ecuación estructural: es un conjunto de ecuaciones lineales de la forma $x_i = f_i(pa_i, u_i)\ i = 1,...,n$ donde cada pa_i son los padres de x_i para un conjunto de variables X y donde el u_i representa el error (o "perturbaciones") debido a factores omitidos.

Un conjunto de ecuaciones descritas de esta forma, en el cual cada ecuación representa un mecanismo autónomo es llamado un modelo de ecuaciones estructurales (Structural Equation Model, SEM). Cuando cada mecanismo determina el valor de únicamente una variable distinta, (llamada en ocasiones variable dependiente) el modelo resultante se denomina estructura causal.

Estructura Causal. La estructura causal de un conjunto de variables V es un grafo dirigido acíclico (DAG) en el que cada nodo corresponde a un elemento distinto en V, y cada enlace representa una relación funcional directa entre las variables. La estructura causal sirve para diseñar el modelo causal, ya que es una especificación precisa de cómo cada variable es influenciada por sus padres en el DAG, en un modelo de ecuaciones estructurales $V = X$.

Modelo Causal. Un modelo causal es un par $M = \langle D, \Theta_D \rangle$ que consiste en una estructura causal D y un conjunto de parámetros Θ_D compatibles con D. Los parámetros Θ_D asignan una estructura causal a cada variable X en D.

Donde cada variable es independiente de todas aquellas que no sean sus descendientes, y a la vez condicional a sus padres. Esta condición es una guía para decidir cuándo se han incluido todas las causas relevantes a la variable X_i. El modelo M es formado define la distribución de probabilidad $P(M)$ sobre todas las variables del sistema.

3.1.4 CREACIÓN DE MODELOS CAUSALES

Etapas del modelado causal

El modelado causal tiene cuatro etapas principales: especificación, estimación, interpretación y evaluación [Cohen95]. La primera consiste en indicar cuáles variables son causas y cuáles son efectos, la segunda involucra aplicar un método computacional para determinar la intensidad de las relaciones causales encontradas, en la tercera se analizan e interpretan los resultados, y en la cuarta se prueba si el modelo predice con exactitud.

Durante la fase de especificación se presentan dos problemas fundamentales para el modelado causal, la presencia de factores no controlados y la especificación del orden causal. Los factores no controlados son aquellos de los cuales el investigador no se ha percatado y por consecuencia no ha medido. El orden causal significa encontrar la dirección correcta de la relación existente entre dos variables.

Una forma de resolver problemas asociados a la presencia de factores no controlados es realizando un diseño de experimentos controlado en el cual el investigador puede intervenir manipulando los valores de las variables y registrando los efectos respecto a las demás variables [Montgomery04]. Logrando de éste modo, evidenciar si las relaciones que supone son verdaderas o no. Este enfoque es difícil de aplicar en el modelado causal de algoritmos metaheurísticos, debido a los parámetros asociados son de naturaleza multifactorial.

En la fase de estimación se calculan las magnitudes de las relaciones causales encontradas. El método utilizado para estimar un modelo causal depende de la naturaleza de los datos que se están analizando. Si se utilizan datos discretos la tarea es encontrar las distribuciones de probabilidad asociadas a cada nodo [Heckerman95],

cuando se tratan datos continuos, se califica el modelo mediante las ecuaciones estructurales asociadas al mismo.

Para la interpretación del modelo se analizan las relaciones que se consideran más fuertes tomando en cuenta su magnitud y se da una explicación en base a la información que cada variable representa. Durante esta etapa es necesario que el investigador considere el conocimiento previo del dominio del problema e interprete de manera objetiva los resultados obtenidos.

Algoritmos para la creación de grafos causales

La mayoría de los métodos utilizados para crear modelos causales están basados en pruebas de independencia condicional, ésta define un conjunto de restricciones que deben ser satisfechas para inducir la estructura causal.

Existe independencia estadística cuando la probabilidad de un evento no depende o se ve afectada por la presencia de otro. Dos variables X y Y son condicionalmente independientes dada una tercera variable W si para cualquier conjunto medible S de posibles valores de W, X y Y son condicionalmente dependientes dado el evento $W \in S$. Si se conoce el estado de W , el conocimiento del estado de Y es irrelevante para saber algo de X.

Dos algoritmos diseñados para crear modelos causales a través pruebas de independencia condicional: PC (las iniciales del primer nombre de sus autores: Peter Spirtes y Clark Glymour [Spirtes01] y Fast Causal Inference (FCI) [Spirtes00]. A continuación se presenta un bosquejo del proceso del algoritmo PC.

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

a. El grafo causal en la población es acíclico
b. El grafo causal en la población es confiable
c. El conjunto de variables causales es causalmente suficiente

El algoritmo PC consiste en dos etapas principales:

1. Eliminación de aristas. Este proceso inicia con la construcción de un grafo completo que relaciona a todas las variables de entrada. Posteriormente en base a pruebas de dependencia condicional se van eliminando aquellas aristas entre variables que son condicionalmente independientes.
2. Orientación estadística de aristas. En esta etapa se crea la orientación del grafo mediante operadores condicionales.

Dos algoritmos alternativos son FBD y FTC [Cohen95], basados en ecuaciones de regresión, estos pueden ser aplicados solo datos continuos, y han demostrado buen desempeño comparados con PC e FCI. El desempeño de FBD es muy similar a PC, pero tiene la ventaja de ser un algoritmo de tiempo polinomial (PC es de tiempo exponencial).

Existen herramientas computacionales disponibles para la creación de modelos causales, como Hugin Tool [Madsen05], Netica [Norsys06] y TETRAD [Carnegie06]. Además de recursos desarrollados para la enseñanza de modelos causales como el laboratorio Causality Lab de la Universidad de Carnegie-Mellon [Carnegie06].

3.1.5 INTERPRETACIÓN DE MODELOS CAUSALES

El análisis causal trata de inferir relaciones entre variables o encontrar evidencia de una relación, pero se enfoca preferentemente a inferir aspectos que se refieren al proceso de la generación de los datos. Esta capacidad incluye predecir los efectos de las intervenciones o cambios espontáneos, identificar las causas de los eventos reportados y evaluar la responsabilidad y atribución de los mismos para la ocurrencia de un evento.

Las expresiones causales pueden, en principio, ser clasificadas como generales y particulares. Sin embargo, aun hay intensos debates respecto a esta clasificación, especialmente en el área de filosofía. Las expresiones del tipo acción-efecto corresponden a expresiones causales generales, mientras que aquellas que se refieren a un efecto a partir de una acción particular corresponden a expresiones causales particulares.

En el contexto de las explicaciones generadas por una máquina, la clasificación anterior tiene significado cognitivo y computacional. En el aspecto cognitivo se refieren al conocimiento que podemos obtener de dichas expresiones, como en el caso de las expresiones genéricas que reflejan el conocimiento obtenido. Sobre el factor computacional se puede indicar que las causas particulares requieren un mayor esfuerzo computacional que las generales, debido a que demandan especificaciones más detalladas.

Para interpretar un modelo causal, en el caso de causas generales, es necesario contar con conocimiento del dominio del problema y analizar las relaciones existentes en el grafo. Las relaciones causales pueden ser probadas como una guía para realizar una acción en presencia de evidencia.

Sin embargo, antes de confiar en una hipótesis propuesta, la habilidad científica debe intentar articular e investigar cada alternativa seria ya que una objeción a los modelos causales en cualquier disciplina es que son arbitrariamente seleccionados.

Algunos investigadores intentan de muchas formas extraer la información causal a partir de las observaciones y hacer las predicciones en base a esa información. Es tan difícil hacer tales inferencias correctamente que algunos críticos han concluido que es imposible desde el principio. Si se aplica el análisis causal a un conjunto de datos y si los resultados son ambiguos o poco informativos, entonces se puede presentar alguno de los siguientes casos:

a. El modelo no contiene ningún camino causal que conecte todas las variables relacionadas propuestas.
b. El modelo generado tiene un número grande de alternativas
c. El rendimiento conecta directamente a cada par de variables o a ninguno

Estos casos indeseables surgen porque las pruebas de dependencia condicional que se usan no son óptimas y son insuficientes para hacer distinciones o porque los algoritmos causales no encuentran soporte para la toma de decisiones acerca de la independencia o dependencia de las variables. Sin embargo, estas dificultades nos indican que hay algo malo con las suposiciones usadas para modelar y que ninguna inferencia fiable puede hacerse de la muestra bajo las suposiciones modeladas impuestas por el usuario.

Si el tamaño de la muestra es muy pequeño para las pruebas estadísticas no se puede probar la variación de los datos para poder tener mejores alternativa en la suposición de linealidad, de normalidad. En estos casos los rendimientos indeseables son una guía real de la verdadera naturaleza de los datos.

3.2 TRABAJOS RELACIONADOS

3.2.1 APLICACIÓN DE MODELOS CAUSALES EN MINERÍA DE DATOS

Los modelos causales son versiones simplificadas de las relaciones existentes en sistemas complejos. En la minería de datos se utilizan principalmente en el proceso de selección de características, debido a que el modelo reduce la cantidad de variables y conserva solo aquellas que muestren relaciones causales directas entre sí. Así como también en tareas de predicción, por la posibilidad de calcular el resultado y efecto de una intervención. Enseguida se refieren brevemente algunos trabajos relacionados.

En [Esposito00] se presenta un trabajo en el que utiliza algoritmos de aprendizaje inductivo y técnicas de inferencia causal para descubrir reglas causales en los atributos

de bases de datos relacionales. El resultado de este trabajo en un sistema llamado CAUDISCO (CAUsal DISCOvery). En el cual el proceso de descubrimiento causal consta de dos fases: inferir la estructura causal de los datos y utilizar el algoritmo de clasificación C4.5 [Quinlan93] para generar un conjunto de reglas para cada dependencia causal relevante. Este trabajo presenta una nueva perspectiva, ya que las reglas de predicción generadas por C4.5 solo tienen una semántica vinculada lógicamente, mientras que las generadas por el sistema mencionado tienen una semántica causal.

En [Lemeire04] se presenta el modelado causal aplicado al análisis del desempeño de un algoritmo paralelo, con la finalidad de detectar las principales causas que originan anomalías observadas en el proceso de comunicación durante varias corridas del algoritmo, en particular para analizar el desempeño del algoritmo Quicksort. El modelo causal se generó aplicando el algoritmo PC de TETRAD a variables relacionadas con el desempeño del algoritmo mencionado, como lo son: tamaño de la entrada, tipo de dato, orden inicial, número de comparaciones e intercambios, entre otros. Como resultado se obtuvieron reglas causales para predecir el tiempo de ejecución de una aplicación en un sistema desconocido, sin embargo se tuvo la limitante de que esto sólo funciona para muestras pequeñas.

3.2.2 ENFOQUES PARA LA CREACIÓN DE MODELOS CAUSALES

Actualmente existen numerosas investigaciones que se enfocan a la construcción de modelos causales para dar explicación a las relaciones encontradas en diversos problemas de investigación.

Algunos enfoques sugeridos por la literatura para construir modelos causales son: funciones de regresión [Cohen95], aprendizaje basado en reglas [Esposito00], redes bayesianas [Pearl04, Spirtes00], aprendizaje basado en explicaciones [Russel04] y

programación lógica inductiva [Russel04]. En la Figura 3.1 se muestra una visión general de estos enfoques.

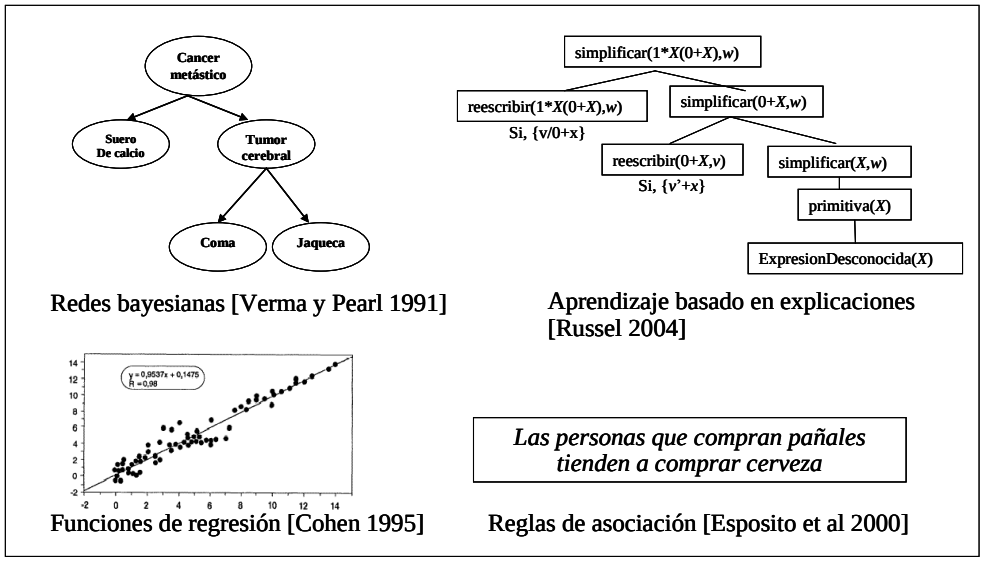


Figura 3.1. Enfoques para la Creación de Modelos Causales

En [Pearl03] se presenta una revisión acerca de los métodos relacionados con la inferencia causal. Presenta una metodología para generar modelos causales a través de representaciones graficas (Grafos Cíclicos Dirigidos, DAG). Muestra métodos prácticos para encontrar relaciones causales en los datos, derivar relaciones causales a través de la combinación de datos y conocimiento, predecir los efectos de acciones y guías para evaluar las explicaciones con sustento causal. Este autor propone el algoritmo IC (Inductive Causation) que esta basado en pruebas de independencia condicional a través de los datos.

En [Spirtes01] se propone el algoritmo PC como herramienta para crear modelos causales para datos causalmente suficientes. En [Spirtes00] se presenta un algoritmo para crear modelos causales llamado The Fast Causal Inference, mismo que busca

características comunes a conjuntos de observaciones equivalentes de grafos causales directos acíclicos. Este algoritmo esta indicado para cuando se cree que los datos que se utilizan no son causalmente suficientes.

En [Maes05] se presenta un paradigma para tratar los modelos causales en redes con configuración multiagente. El autor presenta un algoritmo para la identificación de los efectos causales en contextos en los que el agente no tiene acceso completo a todo el dominio. El método que presenta para encontrar los modelos causales es representar en un grafo dirigido las relaciones existentes entre las suposiciones teóricas que son suficientes para calcular la intervención deseada de las variables.

3.2.3 ANÁLISIS COMPARATIVO

En la Tabla 3.1 se presenta un análisis comparativo de características importantes que presentan los trabajos relacionados con la investigación propuesta. La primera columna, se refiere a si el enfoque presentado está aplicado al análisis de algoritmos, la segunda columna indica si el trabajo incluye un modelado de las características del problema o del algoritmo.

Tabla 3.1 Comparación de trabajos relacionados con el uso de modelos causales en el área de computación

Trabajo	Aplicado al análisis de algoritmos	Modelado de Características	Evalúa el conocimiento adquirido	Utiliza técnicas de aprendizaje	Incorpora explicaciones formales del comportamiento observado
Maes 2005	✓		✓		
Lemeire y Dirkx 2004	✓	✓			
Esposito et al 2000			✓	✓	
El trabajo propuesto	✓	✓	✓	✓	✓

La tercera columna indaga acerca del conocimiento adquirido, si es evaluado mediante algún mecanismo específico. La cuarta indica si se utiliza un enfoque de aprendizaje automático y en la columna 5 se menciona si se incorporan explicaciones formales del comportamiento observado.

Como puede observarse, los trabajos analizados no incluyen todas las características mencionadas. El trabajo propuesto incorpora una característica adicional que es proporcionar explicaciones formales del comportamiento observado, esto puede significar una nueva perspectiva para la aplicación de los modelos causales al área de computación

3.2.4 HERRAMIENTA TETRAD PARA LA CREACIÓN DE MODELOS CAUSALES

TETRAD [Carnegie06] es un software gratuito desarrollado con el objetivo de crear, simular datos, estimar, probar, predecir y buscar modelos estadísticamente causales. Ofrece métodos para descubrimiento causal en una interfaz amigable y sencilla de usar.

Si los datos que se van utilizar provienen de un tamaño de muestra lo suficientemente grande y las suposiciones de normalidad están razonablemente satisfechas, TETRAD puede ayudar con los siguientes problemas entre otros:

- Reconocer cuándo un conjunto de datos proporciona poca o ninguna información sobre los procesos subyacentes.
- Encontrar las alternativas estadísticamente equivalentes a un modelo dado.
- Encontrar alternativas que expliquen adecuadamente los datos del modelo dado.
- Seleccionar las variables que influyen directamente en una variable del resultado.
- Reducir el número de variables requeridas para la predicción.
- Descubrir la existencia de variables ocultas (variables latentes).
- Encontrar la ecuación estructural planteada con las variables latentes.

- Reespecificar modelos utilizando el modelo de ecuaciones estructurales obtenido.
- Construir una red bayesiana que funcione como un sistema especialista para una base de datos.
- Usar una red bayesiana para clasificar y predecir.
- Crear datos simulados generados a partir de los modelos causales.

Si el propósito del análisis de los datos es estimar las influencias causales individuales que cada variable ejerce en el resultado, entonces los procedimientos que aplica TETRAD son teóricamente inestables. Dependiendo de la verdadera estructura causal y qué variables han sido modeladas, es posible identificar qué variables son las causas directas y cuales son los efectos directos.

Así, cuando las suposiciones referentes a la distribución de los datos están justificadas, TETRAD puede usarse para ayudar a la selección de relaciones causales para después estimar la influencia precisa con cualquier paquete de datos.

Interfaz gráfica de TETRAD

En la ventana principal de TETRAD se incluyen las herramientas principales para el modelado causal, que son: crear y simular grafos causales, búsqueda de estructuras causales, estimadores del grafo causal y análisis de regresión, entre otros.

La herramienta búsqueda de estructuras causales (search) cuenta con varias implementaciones de algoritmos de inferencia causal para generar grafos que representan el modelo causal como los son: PC, PCD y FCI entre otros. Estos algoritmos de inferencia causal están basados principalmente en pruebas de independencia condicional realizadas a partir de la matriz de varianzas-covarianzas. Los algoritmos pueden ser aplicados tanto a datos continuos como discretos.

En un grafo obtenido por TETRAD la orientación de las relaciones encontradas depende de qué tan eficiente fue el algoritmo de inferencia para encontrar las relaciones causales. A continuación se describen las posibles orientaciones que pueden obtenerse en el grafo causal.

a) $\rightarrow$ Flecha sencilla. Indica una causa genuina directa
b) $\leftrightarrow$ Flecha doble. Indica la presencia de una causa latente común entre dos variables
c) o$\rightarrow$Flecha sencilla con círculo en un extremo. Indica la incapacidad de TETRAD para deducir si existe una influencia directa entre dos variables ó si existe una causa latente entre ambas.
d) o–o Arista con círculos en ambos extremos. Indica la incapacidad de TETRAD para deducir si hay una influencia directa entre dos variables, y si fuera así, cual sería la dirección, o la causa latente entre ellas.

El modelo causal usando TETRAD se construye en tres etapas: la primera la creación de un grafo causal que especifica relaciones causales hipotéticas entre las variables, la segunda es la especificación de la familia de distribuciones de probabilidad asociadas al grafo y los parámetros asociados con al modelo grafico, y por último una especificación de los valores numéricos de esos parámetros.

Capítulo 4

PROPUESTA DE SOLUCIÓN

En este capítulo se presenta el desarrollo sistemático de una metodología para la creación de modelos causales del desempeño algorítmico. Se propone que ésta sea utilizada como una guía para el análisis del comportamiento de algoritmos metaheurísticos.

El análisis propuesto incorpora la aplicación de un método formal para encontrar y explicar las relaciones entre los factores que definen el comportamiento de un algoritmo. Las explicaciones obtenidas al aplicar esta metodología permiten incrementar el conocimiento que se tiene sobre el comportamiento de los algoritmos. Este conocimiento puede ser utilizado, entre otras aplicaciones, para el diseño de algoritmos adaptados al problema.

Los modelos creados con este proceso incorporan lógica causal, lo cual los hace menos susceptibles al sobre-ajuste, y por tanto más robustos al ser aplicados a tareas de predicción que los métodos que incorporan sólo lógica semántica [Esposito00].

4.1 METODOLOGIA PROPUESTA PARA EXPLICAR EL DESEMPEÑO DE ALGORITMOS METAHEURISTICOS

La Figura 4.1 presenta la metodología propuesta para explicar el desempeño de algoritmo metaheurísticos en problemas de distribución de objetos

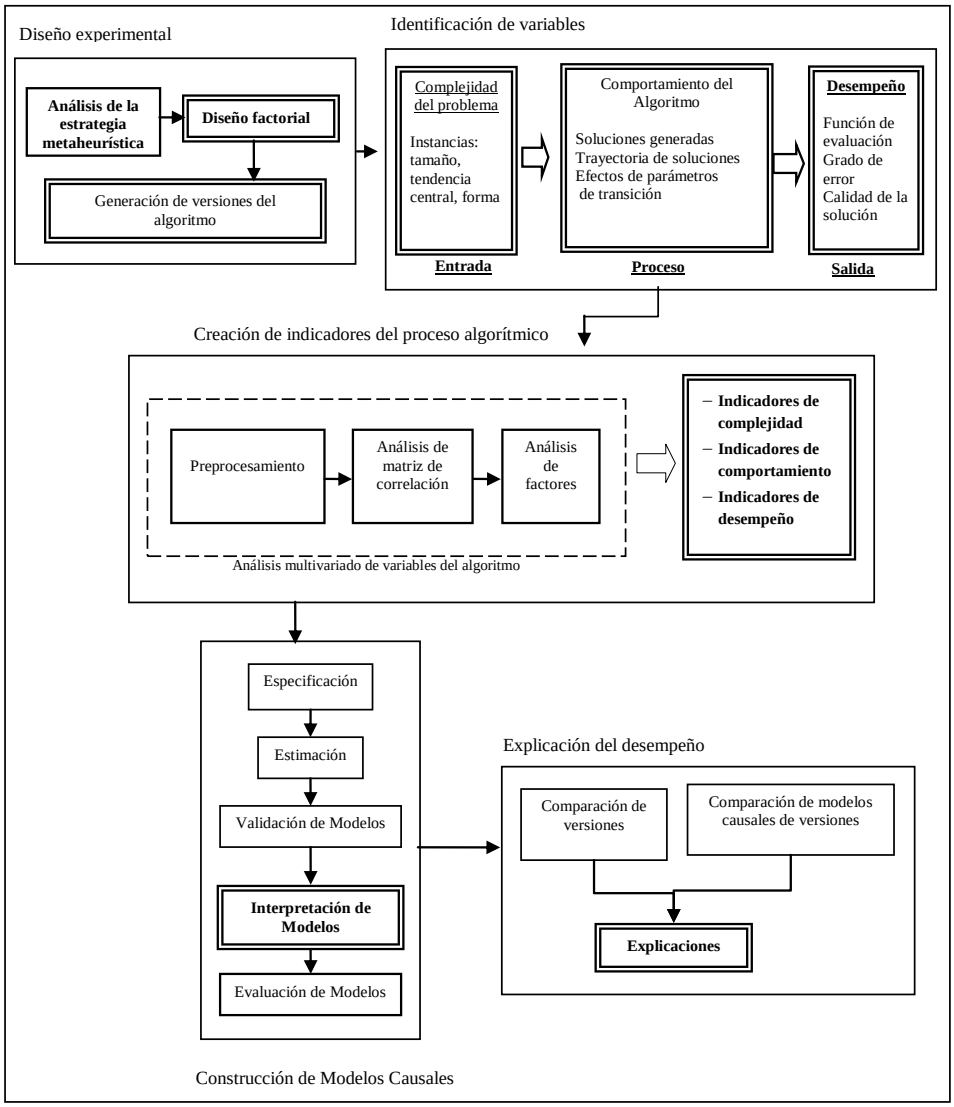


Figura 4.1 Esquema de la metodología propuesta

4.2 ETAPA 1. DISEÑO EXPERIMENTAL PARA ANALIZAR EL DESEMPEÑO ALGORITMICO

Las siguientes secciones proporcionan una descripción de esta etapa que consiste básicamente en analizar el algoritmo e identificar un diseño que permita analizar cambios en su estructura.

4.2.1 ANÁLISIS DE LA ESTRATEGIA METAHEURÍSTICA

En este paso se identifican los elementos principales que conforman la estrategia metaheurística analizada, como son: los criterios para generar la solución inicial, los operadores que definen la vecindad, las reglas que definen las transiciones en el proceso algorítmico y los criterios de paro (ver sección 2.4.1).

4.2.2 DISEÑO FACTORIAL PARA ANALIZAR LA ESTRUCTURA DEL ALGORITMO

Para comprobar la suposición de relaciones entre los elementos que se sospecha influyen en el desempeño del algoritmo, se propone realizar un diseño de experimentos factorial [Montgomery04]. Este tipo de diseños son utilizados para identificar los efectos de las interacciones de un conjunto de factores relacionados con una variable respuesta.

Un diseño experimental factorial completo consiste en evaluar todas las posibles interacciones entre los factores que se analizan. Frecuentemente, los factores tienen varios valores posibles, llamados niveles. En el análisis se consideran todas las combinaciones posibles entre un numero determinado de niveles del factor.

El efecto de un factor se define como el cambio que se produce en la respuesta o medida de desempeño producido por un cambio en el nivel del factor. Cuando la respuesta entre niveles de un factor no es la misma, existe una interacción entre factores.

En esta investigación el objetivo es identificar si existe una relación entre los elementos que definen una estrategia metaheurística y si ésta existe, evaluar los efectos de las interacciones.

Los algoritmos metaheurísticos tienen varios parámetros asociados a los elementos que definen la estrategia. Por ejemplo, para el algoritmo Búsqueda Tabú (ver sección 2.4.2) las reglas de transición del proceso están controladas por los parámetros: tenencia (tiempo que la solución es considerada tabú) y tamaño de la lista tabú, entre otros.

El problema, para fines prácticos, es que deben probarse todas las combinaciones entre parámetros. Esto implica ejecutar el algoritmo con cada combinación de parámetros posible y lleva a una experimentación muy costosa, en términos de tiempo y esfuerzo computacional.

Debido a esta situación, este trabajo sugiere analizar solo los aspectos siguientes: criterios para generar la solución inicial, operadores que definen la vecindad y los parámetros asociados a las reglas de transición.

Estos según la literatura [Fonlup97], son los que han mostrado mayor impacto en el desempeño del algoritmo. También se sugiere evaluar al menos dos niveles para los parámetros asociados al aspecto medido. La Tabla 4.1 muestra el diseño factorial descrito.

Tabla 4.1 Diseño factorial de elementos que afectan una estrategia metaheurística

Factor	Niveles	
Solución inicial	Aleatoria	Heurística
Operadores de búsqueda	Selección aleatoria entre operadores	Operador único
Parámetros asociados a las reglas de transición	Valor fijo	Auto-configuración estática

4.2.3 APLICACIÓN DEL DISEÑO FACTORIAL: GENERACIÓN DE VERSIONES DEL ALGORITMO

Utilizando el diseño factorial se establecen configuraciones para el análisis del algoritmo, lo que genera varias versiones del mismo. La Tabla 4.2 presenta el diseño factorial completo utilizado en este trabajo.

Las versiones que resultan, deben ser analizadas individualmente para identificar si el cambio de estrategia impacta en el desempeño que obtiene el algoritmo para una misma entrada. Se sugiere configurar las versiones variando un nivel de factor a la vez.

Por ejemplo, en la Tabla 4.2, las versiones 1 y 2 presentan la misma configuración en los parámetros de control y en la estrategia para obtener la solución inicial; el cambio se refiere la selección de los operadores aplicados en la búsqueda local de soluciones, que puede ser aleatoria o fija.

Tabla 4.2 Configuraciones generadas a partir del diseño factorial

Configuraciones						
Versión	Parámetros de control		Solución inicial		Operadores de Búsqueda	
	Valor Fijo	Auto-configuración Estática	Aleatoria	Heurística	Operadores al azar	Un sólo operador
1	Si	No	Si	No	Si	No
2	Si	No	No	No	No	Si
3	Si	No	No	Si	Si	No
4	Si	No	No	Si	No	Si
5	No	Si	Si	No	Si	No
6	No	Si	Si	No	No	Si
7	No	Si	No	Si	No	Si
8	No	Si	No	Si	Si	No

4.3 ETAPA 2. CREACIÓN DE MODELOS CAUSALES DEL DESEMPEÑO ALGORITMICO

A continuación se presenta el procedimiento propuesto para crear los modelos causales para analizar el desempeño algorítmico. Este procedimiento esta formado por varias etapas. La primera consiste en analizar la estructura interna del algoritmo con el objetivo de caracterizar su comportamiento mediante la generación de indicadores.

En la segunda se construyen los modelos causales utilizando los indicadores generados. La siguiente etapa consiste en interpretar el modelo encontrado a través de su estimación y del conocimiento previo que se tenga del dominio de aplicación y la última en evaluar el modelo obtenido.

4.3.1 IDENTIFICACIÓN DE VARIABLES QUE INFLUYEN EN EL DESEMPEÑO DEL ALGORITMO

En esta etapa, el objetivo principal es identificar aspectos de la implementación del algoritmo que sean factibles de medición y que tentativamente proporcionen información útil para describir el desempeño del mismo.

Este análisis tiene que ver con la implementación del algoritmo ya que los algoritmos metaheurísticos, al ser multipropósito, tienen muchas variantes que se ajustan al problema que resuelven. Por esta razón, la medición de variables esta ligada directamente a la estrategia utilizada.

Categorización de variables

Debido a que la conducta del algoritmo puede ser diferente con cada posible entrada, para un estudio completo se propone ver el algoritmo como un flujo de información;

cuyos elementos son: entrada (casos), proceso (comportamiento del algoritmo) y salida (eficiencia obtenida). En la Figura 4.2 se ejemplifica este enfoque.

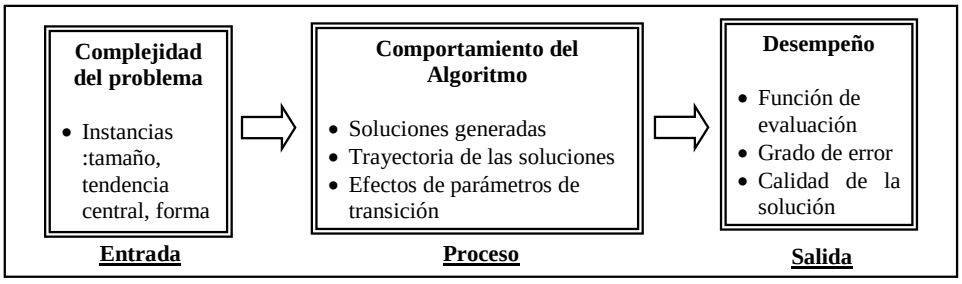


Figura 4.2 Elementos que influyen en la conducta de un proceso algorítmico

En la entrada, los casos proporcionan información acerca de la complejidad del problema; en el proceso los elementos principales son aquellos que proporcionan ideas acerca del comportamiento del algoritmo.

Estos pueden ser: las soluciones que se generan, la trayectoria que describen las mismas y el efecto de los parámetros de transición; y en la salida, son indicadores importantes del desempeño la función de evaluación utilizada, el grado de error obtenido y la calidad de la solución encontrada.

Identificación de variables en cada categoría

Una vez identificados los elementos que influyen a la entrada, el proceso y la salida, se analiza qué aspectos es posible medir en cada categoría. Se debe estudiar la implementación utilizada y seleccionar variables que revelen tendencias importantes.

En la Figura 4.3 se muestran para cada categoría, que aspectos se propone medir para analizar el desempeño de algoritmos en problemas de distribución de objetos.

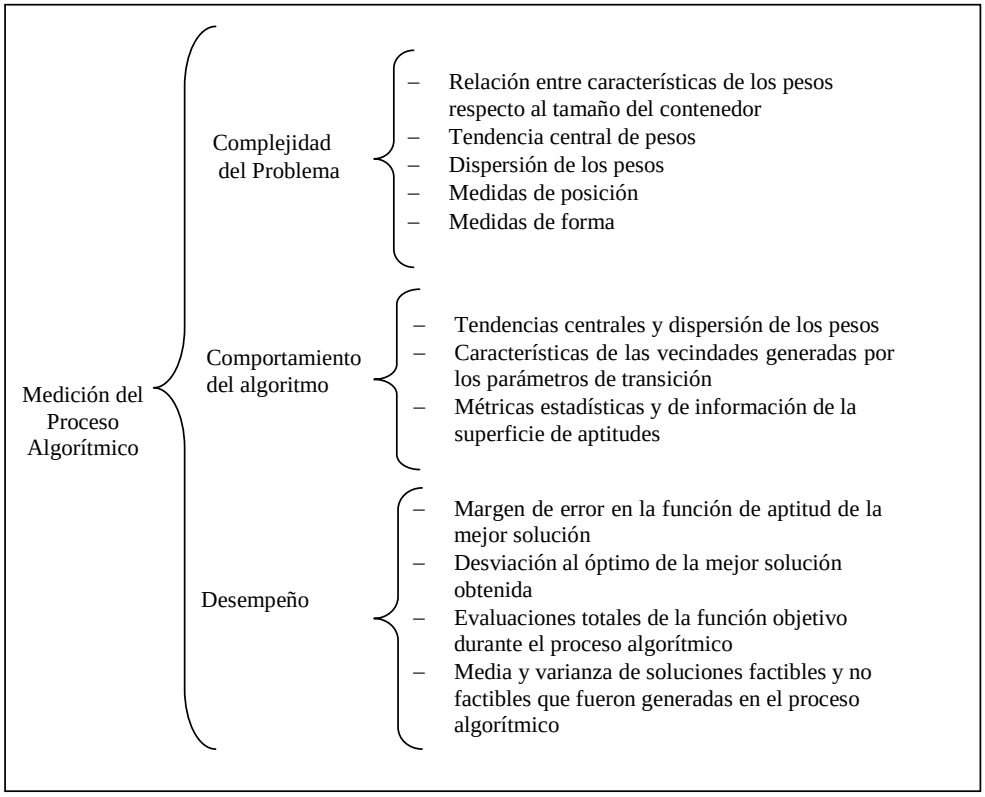


Figura 4.3 Variables que miden el proceso algorítmico

4.3.2 CREACIÓN DE INDICADORES DE COMPLEJIDAD DEL PROBLEMA, COMPORTAMIENTO Y DESEMPEÑO ALGORÍTMICO

En este paso se seleccionan, a partir de las variables creadas en el paso anterior, subconjuntos de variables que se convertirán en los indicadores asociados a la complejidad del problema, el comportamiento y desempeño del algoritmo.

Para crear los indicadores se proponen tres pasos: preprocesamiento de datos, análisis de la matriz de correlación y análisis de factores. La Figura 4.4 muestra el proceso de creación de los indicadores a partir de las variables originales.

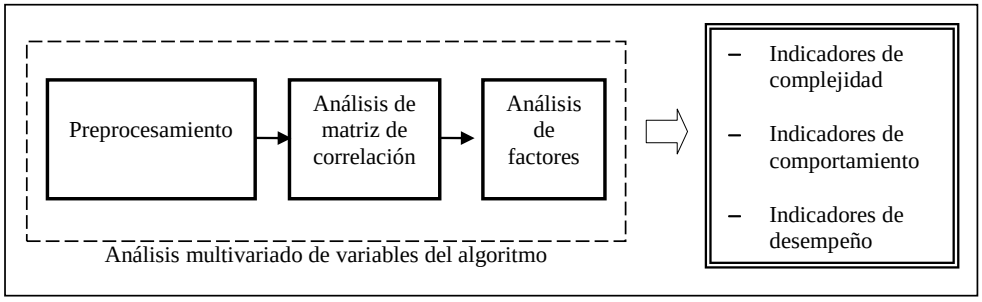


Figura 4.4 Proceso de creación de indicadores

Procesamiento

En este paso se efectúa la preparación y transformaciones necesarias para que los datos estén listos para análisis posteriores. Es importante manejar una misma magnitud para los datos de entrada, para evitar problemas asociados como el sobre-ajuste y la predominación de las variables con mayor magnitud [Johnson02]. Para esto se realiza una estandarización de datos, definiendo un mismo rango para todas las variables.

Análisis de la matriz de correlación

Este análisis permite identificar la existencia de anomalías en los datos, por lo que es importante realizarlo antes de aplicar cualquier análisis estadístico que tenga que ver con correlación. El caso mas preocupante es la presencia de colinealidad en los datos (relaciones lineales entre variables que pueden producir correlación excesiva).

Si la matriz de correlación tiene colinealidad es posible que los análisis estadísticos basados en ésta no tengan sentido o el resultado sea erróneo [Johnson00]. Una solución simple para este problema, en caso de tener variables con valor de correlación perfecta o muy alta, es seleccionar una sola y descartar las demás.

Análisis de factores

Una forma de reducir la cantidad de variables es analizar los datos con un método de reducción de variables para identificar si existe un conjunto simplificado que represente la misma varianza y características subyacentes que el conjunto original. La reducción de variables ayuda a evitar problemas de colinealidad [EMVI07].

El análisis de factores explica la estructura de correlación en las variables estudiadas a través de la obtención de un subconjunto de nuevas variables independientes no correlacionadas llamadas factores. Las nuevas variables son una combinación lineal de las variables originales. El anexo B presenta una descripción de este procedimiento.

Al aplicar el análisis de factores, es conveniente verificar la existencia de un patrón de factores entre las versiones analizadas del algoritmo. Esto facilitara las comparaciones entre modelos y ayudara a identificar las diferencias entre configuraciones.

4.3.3 CONSTRUCCIÓN DE MODELOS CAUSALES

La construcción del modelo causal es la parte medular del modelado causal. Una vez identificados los indicadores que participaran en el modelo éstas se analizan con un método de inferencia causal para determinar las relaciones existentes entre ellas.

La creación del modelo se hace en dos pasos que son descritos en las secciones siguientes.

4.3.3.1 ESPECIFICACIÓN DEL ORDEN CAUSAL

Especificar el orden causal es una tarea que consiste en identificar las variables que se encuentran relacionadas y la dirección que tiene esta relación. La tarea se complica

cuando el conjunto de variables que participan en el modelo están influenciadas por un factor oculto, llamado variable latente, que el investigador no ha contemplado.

La creación del modelo implica un esfuerzo computacional considerable, ya que es necesario encontrar, en el espacio de búsqueda asociado de modelos posibles, el modelo que mejor ajuste a los datos. Éste es un problema combinatorio y crece exponencialmente con el número de variables presentes en el modelo.

Planteamiento de hipótesis

Generalmente, cuando se desea modelar un fenómeno observado, se tiene cierto conocimiento del dominio de la aplicación que se esta observando. En el modelado causal, este conocimiento puede ser incluido bajo la forma de hipótesis acerca de cómo serán las relaciones que se espera encontrar.

Basándose en el conocimiento obtenido de trabajos relacionados y en la experiencia adquirida durante el desarrollo del trabajo de investigación, se propone probar las siguientes relaciones hipotéticas entre los indicadores que se considera afectan el desempeño algorítmico.

1. Indicadores de comportamiento ← Indicadores de complejidad
   Los indicadores referentes a la complejidad del problema influyen directamente en los factores relacionados con el comportamiento del algoritmo [Kallel01]. Esto debido a que la dificultad para resolver el problema tiene que ver en el desempeño de un algoritmo, si el espacio de soluciones del problema es complejo, es de esperarse que la trayectoria que genere el algoritmo refleje esta situación.
2. Indicadores de desempeño←Indicadores de comportamiento
   Esta hipótesis tiene que ver con la idea de que el operador de búsqueda define la trayectoria [Jones95a], misma que al final se traduce en el desempeño del

algoritmo. Es de esperarse que cuando el operador utilizado en la búsqueda es adecuado el algoritmo muestre buen desempeño.

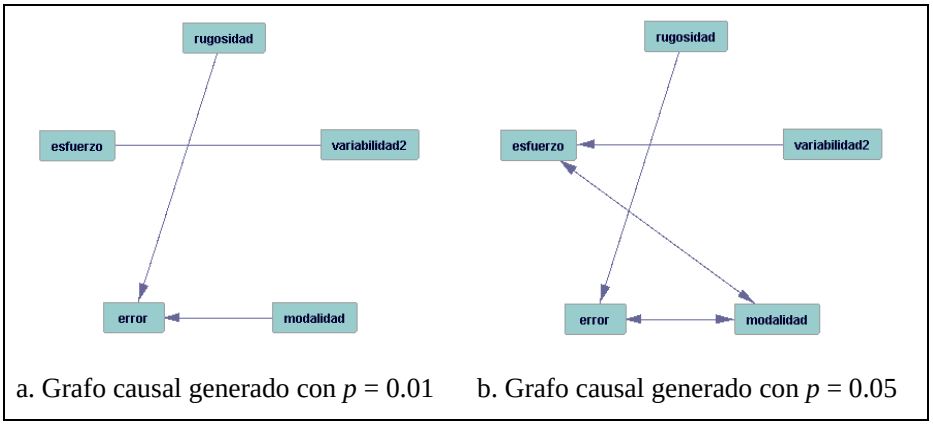
Creación de grafos causales

Un grafo causal es una representación gráfica de las relaciones causales que se identifican para el conjunto de variables analizadas. En este trabajo se optó por utilizar la herramienta computacional TETRAD [Carnegie06] reconocida por la literatura como una opción aconsejada para la creación de modelos causales [Bowes04]. En TETRAD existen varios algoritmos disponibles para la creación de grafos causales.

Para crear los modelos del desempeño de algoritmos se seleccionó como algoritmo de inferencia al algoritmo PC [Spirtes01]. Éste tolera la existencia de variables latentes, situación que se presenta frecuentemente durante la obtención de un grafo causal. Por esta razón, se considera que PC es una buena opción para generar los modelos asociados a las hipótesis presentadas en la sección anterior.

El grafo causal obtenido con TETRAD puede variar notoriamente dependiendo del nivel de significación *p* seleccionado para realizar las pruebas de independencia entre variables. Es por esto que se recomienda hacer pruebas a diferentes niveles de significación. Debe considerarse que si se utiliza un valor de significación muy pequeño es probable que se rechacen relaciones causales débiles pero genuinas, y que al contrario, un valor grande introduzca relaciones engañosas [Glymour&Cooper99].

La Figura 4.5 muestra un ejemplo de grafos causales obtenidos por TETRAD, utilizando p=0.01 y p=0.05. Como se observa, un valor mayor para p genera más relaciones causales en el modelo; sin embargo son engañosas, ya que el sentido de la orientación no esta especificado. Por lo que en este caso, es mejor usar un valor menor de p.



a. Grafo causal generado con $p = 0.01$ b. Grafo causal generado con $p = 0.05$

Figura 4.5 Grafos causales obtenidos con TETRAD para los p= 0.01 y p=0.05

4.3.3.2 ESTIMACIÓN DEL ORDEN CAUSAL

En esta etapa se estima la magnitud de las influencias entre las variables observadas. Las influencias en el grafo causal se indican a través de la orientación de las aristas. Por ejemplo, A→B indica la influencia de *A* en *B*.

Un grafo causal indica visualmente la estructura subyacente, pero esta debe ser estimada para conocer la magnitud real de las influencias encontradas. En la literatura especializada existen diversas alternativas para realizar este procedimiento dependiendo si se trata de variables continuas o discretas.

Si las variables medidas tienen intervalos discretos, la magnitud de las relaciones se estima a través de las distribuciones de probabilidad asociadas a los nodos de interés. Las relaciones se estiman de la siguiente manera: P(*variables*|*conjunto_evidencia*).

Por ejemplo, en la Figura 4.5a la magnitud de la relación entre las variables *rugosidad*, *modalidad* y *error* es: P(*error*=valor|*rugosidad*=valor,*modalidad*=valor) para cada uno de los valores posibles de la variable *error*, *rugosidad* y *modalidad*.

Cuando los datos son continuos, la estructura causal es generada a través de un modelo de ecuaciones estructurales, donde cada variable en el modelo es estimada por una expresión lineal que es una suma lineal de sus padres más un término de error.

La expresión asociada es: $t\_error+variable=c_1Padre_1+c_2Padre_2+\ldots+c_nPadre_n$. Para estimar el valor de *variable* es suficiente con calcular la expresión asociada. Como ejemplo, para la relación *rugosidad*, *modalidad* y *error* de la Figura 4.5a la ecuación lineal es error = 0.14 + 0.81rugosidad - 0.76modalidad.

4.3.3.3 VALIDACIÓN DEL MODELO CAUSAL

Para validar si el modelo causal encuentra realmente la estructura causal subyacente en los datos, se utiliza se utiliza una prueba de hipótesis mediante el estadístico χ^2 [Mitchel97]. Esta prueba asume que la estimación ha sido maximizada para las variables medidas. Bajo esta suposición la hipótesis nula se plantea de la siguiente manera:

H_0: La matriz de covarianzas de la población es igual a la matriz de covarianza estimada para las variables medidas.

H_1: La matriz de covarianzas de la población no es igual a la matriz de covarianza estimada para las variables medidas.

Donde las variables medidas están escritas en función de un modelo de parámetros libres (estimados a partir de los datos). Cuando las variables son continuas, estos parámetros son:

- Parámetros no ajustados para cada arista no dirigida (el coeficiente de cada arista)
- La varianza del error considerado para variables exógenas del modelo (causas)
- La covarianza de las variables que conecta (cuando se presentan aristas bidireccionales)

Si la hipótesis se acepta, a un nivel de confianza dado, se puede afirmar que el modelo generado describe adecuadamente la estructura causal de la población que representa.

4.3.3.4 INTERPRETACIÓN DE MODELOS CAUSALES

En esta etapa se analizan las relaciones causales de mayor magnitud para generar explicaciones causales, tomando en cuenta el significado de cada variable y sus interacciones en el contexto del dominio de la aplicación.

Interpretación de relaciones causales para datos discretos

En el caso de datos discretos, se analizan las estimaciones cuyos valores de probabilidad sean mayores. Posteriormente, se explica la relación encontrada asociando los valores que puede tomar cada variable (niveles) y se interpreta tomando en cuenta el significado de la variable en el contexto de la aplicación.

Por ejemplo para las relaciones del la Figura 4.5 entre las variables *rugosidad*, *modalidad* y *error*, si cada una de estas variables tuviera asociados dos niveles: grande, pequeño codificados como 1 y 0. Se evaluaría la distribución de probabilidad asociada, ver Tabla 5.3.

Se identifican los valores mayores de probabilidad, que se muestran sombreados en la tabla, y se interpreta de la siguiente manera: existe mayor error cuando la rugosidad del problema es alta y la modalidad baja, mientras que el error es menor cuando la rugosidad es menor y la modalidad también.

Tabla 4.3 Cálculo de probabilidades para una relación causal

Error	P(*error*|*rugosidad=1, modalidad=1*)	P(*error*|*rugosidad=1, modalidad=0*)	P(*error*|*rugosidad=0, modalidad=1*)	P(*error*|*rugosidad=0, modalidad=0*)
1	0.45	**0.85**	0.77	0.02
0	0.55	0.15	0.23	**0.98**

Interpretación de relaciones causales para datos continuos

Para datos continuos es posible interpretar cada relación de interés analizando la ecuación lineal que la representa donde los coeficientes indican la cantidad y sentido de las contribuciones de los padres en la variable.

Por ejemplo, en la ecuación *error*=0.14+0.81*rugosidad*-0.76*modalidad*, los coeficientes 0.81 y -0.76 indican la cantidad que aumenta o disminuye el *error* por cada incremento de *rugosidad* o *modalidad* en una unidad. También es posible decir que el error aumenta conforme la *rugosidad* aumenta y la modalidad *disminuye*.

4.3.3.5 EVALUACION DE MODELOS CAUSALES

En esta etapa se comprueba si el modelo realiza predicciones confiables para el conjunto de datos que lo originó, así como también para diferentes conjuntos de prueba.

Siguiendo con el ejemplo dado en las secciones anteriores es posible, en el caso discreto predecir el valor de *error* que tendrá una nueva observación calculando, a partir del modelo de probabilidades, el valor de probabilidad asociado a la combinación de valores dados para las variables *rugosidad* y *modalidad*.

Mientras que para el caso continuo la predicción consiste en utilizar la expresión lineal asociada a *error* para calcular su nuevo valor. Finalmente, como en cualquier tarea de clasificación es posible estimar la calidad de la predicción calculando el porcentaje de los datos de prueba que han sido clasificados correctamente o empleando mecanismos de pruebas de hipótesis como χ^2 y la validación cruzada [Michell97].

4.4 ETAPA 3. ANÁLISIS DE MODELOS CAUSALES PARA LA EXPLICACIÓN DEL DESEMPEÑO DE ALGORITMOS METAHEURÍSTICOS

En esta última etapa se analizan los modelos causales creados para cada versión del algoritmo que resultan al aplicar el diseño experimental. El objetivo principal es utilizar el conocimiento que se tiene de la estructura causal asociada a cada estrategia para analizar si un cambio en la estrategia impacta en el desempeño del algoritmo.

Para evaluar si un cambio en la estrategia afecta el desempeño de alguna forma, se realiza una comparación entre las versiones del algoritmo. Con el propósito de identificar las mejores versiones y observar la existencia de algún patrón, que permita explicar el comportamiento del algoritmo.

Para establecer si las relaciones encontradas tienen relación con la estrategia utilizada se sugiere realizar comparaciones entre aquellos pares que varían sólo un nivel a la vez. De esta forma se controla el experimento y se puede aislar el efecto del cambio de un factor a la vez.

Capítulo 5

RESULTADOS EXPERIMENTALES

En este capítulo se presentan los resultados obtenidos en la aplicación de la metodología propuesta. Se describe el ambiente experimental y posteriormente se presenta de forma detallada el modelado causal del algoritmo Búsqueda Tabú.

5.1 AMBIENTE EXPERIMENTAL

La Tabla 5.1 muestra el hardware y software utilizado para la ejecución de los experimentos. Dos servidores fueron utilizados en la ejecución de los algoritmos y cálculo de métricas de complejidad del problema, comportamiento y desempeño.

Tabla 5.1 Hardware y Software utilizado

Hardware	Software
a. Servidor DELL PowerEdge con 4 procesadores Xeon a 3.06 Ghz, RAM de 3.87 GB. Sistema Operativo. Microsoft Windows Server 2003 para pequeños negocios b. Servidor genérico con Procesador Xeon de doble núcleo 3.2GHz de velocidad, 3.87 Gb de RAM, Sistema operativo Windows Vista Business	– BorlandC++ 5.1 – JSDK1.5.0_06 – Matlab 7
a. Maquina genérica de escritorio, CPU Pentium D a 2.8 Ghz, RAM de 512 Mb, Sistema Operativo Microsoft Windows XP Profesional Service Pack 2. b. Maquina de escritório DELL, CPU P4 a 2.4 Ghz, RAM de 1 GB, Sistema Operativo Microsoft Windows XP Professional Service Pack 2.	– Netbeans 5.5 – TETRAD 4.3.8 – Minitab 14 – SPSS 15

Con formato: Portugués (Brasil)

Los equipos de escritorio fueron utilizados para el proceso de análisis de datos y creación de modelos causales.

Descripción de casos Bin Packing

Una instancia de Bin Packing tiene los siguientes parámetros: *n* = número de objetos a distribuir, *c* = capacidad de los contenedores en los que se distribuirán los objetos y *W* = una lista de pesos de los objetos a distribuir.

Descripción de casos de prueba utilizados

Se utilizó un conjunto de casos estándar reconocidos por la comunidad científica disponibles en la librería OR [Beasley06]. La Tabla 5.2 muestra los parámetros utilizados para la generación y notación de estos casos.

Las principales características entre tipos de casos son las diferencias en los rangos que generan la secuencia de pesos. Por ejemplo, objetos que están en rangos muy pequeños, grandes o proporcionales al tamaño del contenedor.

Tabla 5.2. Descripción de casos utilizados

Nombre	*n*	*c*	W	Notación	Casos utilizados
Hard	200	1000	[2000-35000]	Hard1…Hard9	2
$n_x c_y W_z Rv$	x1 = 50, x2=100, x3=200, x4=500	y1=100 y2= 120 y3=150	z1=[1-100] z2=[20-100] z3=[30-100]	*Rv* donde *v=A…T* para los 20 casos de cada clase	96
$n_x W_y B_z R_v$	x1= 50, x2=100, x3=200, x4= 500	100	y1=[c/3], y2=[c/5], y3=[c/7], y4=[c/9] z1=20%, z2=50% , z3=90%	*Rv= A…T* para los 10 casos de cada clase	190
$U_{x\_}v$	x1=120,x2=250, x3=500,x4=1000	150	(20-100)	*v*= 0…19	20
$T_{x\_}v$	x1=60,x2=120, x3=249,x4=501	100	(25, 50) valores reales	*v*= 0…19	20

Del conjunto total disponible se selecciono un tamaño de muestra de 328 casos con una cantidad representativa de cada tipo de caso. En la Tabla 5.2 se muestra, en la columna casos utilizados, el número de casos de cada tipo de que forman la muestra seleccionada.

5.2 APLICACIÓN DEL DISEÑO EXPERIMENTAL PARA ANALIZAR LA ESTRUCTURA DEL ALGORITMO

Se aplicó la metodología propuesta al algoritmo metaheurístico Búsqueda Tabú. Las siguientes secciones presentan de manera detallada el procedimiento realizado.

5.2.1 ANÁLISIS DEL ALGORITMO BÚSQUEDA TABÚ

Para el análisis del algoritmo, el primer paso fue seleccionar una estrategia, que mostrara un buen desempeño en la solución del problema Bin Packing. Debido a que la metaheurística presenta muchas variantes, se realizaron pruebas con diferentes estrategias y se decidió por una de ellas, conocida como Tabú con Umbral de Aceptación [Glover86]. En el anexo A se presenta la documentación de la implementación y desempeño del algoritmo implementado. El paso siguiente fue identificar los elementos principales que integran la estrategia metaheurística, éstos se describen a continuación:

Parámetros de control

a. Tamaño de lista tabú
b. Tiempo de permanencia en la lista tabú
c. Tamaño de la lista de candidatos
d. Valor inicial de umbral de aceptación de soluciones peores
e. Factor de decremento de umbral

Solución inicial

a. Aleatoria. La solución se crea incrementalmente, al inicio se asigna aleatoriamente un objeto al primer contenedor de la solución. Posteriormente de

entre los objetos restantes se crea un grupo de candidatos (aquellos de tamaño menor al espacio disponible en el contenedor actual) y se selecciona aleatoriamente uno de ellos para añadirlo al contenedor actual. Esta acción se realiza hasta que no se encuentren candidatos, en este momento se añade otro contenedor a la solución y se continúa el proceso hasta asignar todos los objetos.

b. Heurística. El procedimiento es similar al anterior, pero se utiliza una estrategia heurística para seleccionar el objeto a asignar de la lista de candidatos. Ésta consiste en establecer el valor de contribución a la función de aptitud de los elementos que se encuentran en la lista de candidatos y seleccionar el que ayude a mejorar la solución obtenida.

Operadores que definen la vecindad

a. Se implementaron los cuatro operadores descritos en el anexo A
b. Se manejan dos estrategias para seleccionar al operador que se aplicará en la generación de la vecindad: un operador fijo usando intercambio 1-0 y selección aleatoria entre los cuatro operadores disponibles.

Reglas de transición

a. No aceptar soluciones que se encuentren en la lista tabú
b. Si una solución cumple su tiempo de permanencia como tabú (tamaño de lista tabú), volver a considerarla para ser seleccionada
c. Si una solución no es tabú, y no mejora a la mejor solución encontrada hasta el momento, considerarla si su diferencia respecto a la solución actual se encuentra en el umbral α

Criterios de parada

a. Obtener la mejor solución teórica
b. Alcanzar el número máximo de iteraciones establecidad
c. Repetir una solución *m* veces

5.2.2 CREACIÓN DE VERSIONES DEL ALGORITMO BÚSQUEDA TABU

Después de identificar los elementos principales de la metaheurística, se aplicó el diseño factorial presentado en la metodología propuesta. Obteniendo de esta forma 8 versiones que involucran cambios entre parámetros de la configuración del algoritmo.

Las versiones generadas se presentan en la Tabla 5.3, en la que se muestran dos opciones por parámetro de control.

Tabla 5.3 Configuración de versiones del Algoritmo Búsqueda Tabú

Configuración						
Versión	Parámetros de control		Solución inicial		Búsqueda	
	Valor	Auto configuración estática	Aleatoria	Heurística	Operadores al azar	Un solo operador
	Tamaño de lista Tabú	Tamaño de lista Tabú				
1	7	--	Si	No	Si	No
2	7	--	Si	No	No	Si
3	7	--	No	Si	Si	No
4	7	--	No	Si	No	Si
5	-	$\sqrt{n}$	Si	No	Si	No
6	--	$\sqrt{n}$	Si	No	No	Si
7	--	$\sqrt{n}$	No	Si	No	Si
8	--	$\sqrt{n}$	No	Si	Si	No

5.3 CREACIÓN DE MODELOS CAUSALES PARA VERSIONES DEL ALGORITMO BÚSQUEDA TABU

Se crearon los modelos causales para cada versión generada, el procedimiento para todos los casos es el mismo y debido a lo extenso del análisis, en las secciones siguientes se presenta solo el correspondiente a la versión 1. El análisis realizado para las versiones restantes se presenta en el anexo D.

5.3.1 IDENTIFICACIÓN DE VARIABLES QUE INFLUYEN EN EL DESEMPEÑO ALGORÍTMICO

Se analizó que aspectos de los propuestos en la sección 4.3.2 era posible medir para el algoritmo. Identificando las estrategias principales que definen a las versiones propuestas en la sección anterior.

Tabla 5.4 Variables propuestas para medir la complejidad del problema

Tipo de Medición	Variables propuestas	Descripción de variables propuestas
Tamaño del problema	*p*	Es la relación entre el tamaño del caso y el caso Bin Packing mas grande que ha sido resuelto
Relaciones entre los pesos de los objetos a distribuir y el tamaño del contenedor	*b*	Proporción de la suma de objetos que es posible asignar a un contenedor
	t	Capacidad ocupada por un objeto promedio
	f	Expresa la cantidad de factores que hay entre los pesos
	d	Dispersión de los pesos a distribuir en relación con la capacidad del contenedor
Medidas de tendencia central de los pesos	*ma*	Media aritmética de los pesos
	mg	Media geométrica de los pesos
	mh	Media armónica de los pesos
	med	Mediana de los pesos
	moda	Moda de los pesos
Dispersión de pesos: rango, error, coeficiente de variación, varianza, error estándar.	*r*	Rango
	dm	Desviación media de los pesos
	var	Varianza de los pesos
	s	Desviación estándar de los pesos
	s2_n1	Cuasivarianza
	s_n1	Cuasidesviación estándar
	cv	Coeficiente de variación
	e	Error estándar
Posición de los pesos	*c4*	Cuartiles
	c10	Deciles
	c100	Percentiles
Forma de la distribución de los pesos	*ap*	Asimetría de Pearson
	amed	Asimetría de Pearson basada en la mediana
	amod	Asimetría de Pearson basada en la moda
	aboy	Asimetría de Bowley
	curtosis	Curtosis de la distribución de los pesos

La Tabla 5.4 presenta un resumen de las variables propuestas para caracterizar la complejidad del problema, las expresiones necesarias para calcular estas variables pueden ser consultadas en [Cruz04, Álvarez06].

Tabla 5.5 Variables propuestas para medir el comportamiento del algoritmo

Tipo de medición	Variables Propuestas	Descripción de variables
Tamaño de la vecindad.	*cl_s*	Tamaño de lista de candidatos
	v_cls	Varianza del tamaño de lista de candidatos
Medidas de tendencia central y dispersión de la función de aptitud de las mejores soluciones obtenidas	*m_b*	Media de las soluciones óptimas
	v_b	Varianza de las soluciones óptimas
Correlación de las soluciones en la trayectoria del algoritmo	*ro*	Coeficiente de auto-correlación
	delta	Longitud de auto-correlación
Información total y parcial de la trayectoria generada por el algoritmo	*H0*	Contenido de información
	M0	Contenido parcial de información
Modalidad de la trayectoria generada por el algoritmo	*e_opt*	Número esperado de óptimos
	h0	Densidad de los valles

Tabla 5.6 Variables Propuestas para medir el Desempeño del Algoritmo

Tipo de medición	Variables Propuestas	Descripción de variables
Desviación del valor óptimo	*desv_opt*	Desviación respecto a la mejor solución teórica
Número de evaluaciones de la función de aptitud	*pasos*	Evaluaciones totales de la función de aptitud normalizado con el número de pesos de la instancia
Valor del mejor valor de aptitud obtenido	*desv_apt*	Error obtenido en la función de aptitud
Tendencia central y dispersión de soluciones generadas por el algoritmo	*p_f*	Promedio de soluciones factibles encontradas
	m_fact	Promedio de función de aptitud de soluciones factibles
	v_fact	Varianza de función de aptitud de factibles
	p_uf	Promedio de soluciones infactibles encontradas
	br	Mejor corrida
Medidas del engaño del problema	*engaño*	Coeficiente de engaño

Con formato: Centrado

En la Tabla 5.6 se presentan las variables propuestas para la caracterización del desempeño obtenido por el algoritmo, estas son en resumen medidas estandarizadas del error y esfuerzo obtenido por el algoritmo.

La Tabla 5.5 presenta las variables propuestas para caracterizar el comportamiento del algoritmo, estas son principalmente medidas de la tendencia central de las soluciones generadas y métricas de la superficie de aptitudes (ver sección 2.5).

5.3.2 CREACIÓN DE INDÍCADORES DE COMPLEJIDAD, COMPORTAMIENTO Y DESEMPEÑO ALGORITMICO

Para esta etapa se aplicó el proceso descrito en la sección 4.2.3 a las versiones creadas para el algoritmo Búsqueda Tabú. Las siguientes secciones presentan el proceso de manera detallada.

Preprocesamiento

Se analizaron las magnitudes de todo el conjunto de variables y se identificaron aquellas variables que requerían ajustes. La estandarización se realizó aplicando el método Min-Max [Mitchell97] para generar valores en el intervalo [0,1]. A continuación se describen los ajustes para cada categoría de variables:

a. Complejidad. Se estandarizaron todas las variables incluidas en esta categoría
b. Comportamiento. En esta caso solo fueron estandarizadas las variables *cls*, *v_cls* y *e_opt*, ya que las demás ya se encontraban en el intervalo 0-1;
c. Desempeño. Se estandarizaron las variables *pasos*, *v_f* , *br* y *engaño*.

Análisis de la matriz de correlación

Se analizó la matriz de correlación de los grupos de variables que miden la complejidad del problema, el comportamiento del algoritmo y el desempeño obtenido. Debido a lo

extenso de la matriz de correlación para las variables de la complejidad del problema, en la Tabla 5.7 se presentan una muestra.

En la tabla se observa que existen variables que tienen correlación perfecta con más de una variable. Esto sugiere un alto nivel de colinealidad en la matriz, implicando alta redundancia entre variables. Por ejemplo, *ma* (media aritmética), *mg* (media geométrica) y *mh* (media armónica) son redundantes, y esto es hasta cierto punto intuitivo, ya que son diferentes formas de encontrar la media del conjunto de observaciones.

Tabla 5.7 Matriz de correlación de variables que miden la complejidad del problema

	med	*moda*	*var*	*s1*
med	**1.0**	**1.0**	**1.0**	**1.0**
moda	**1.0**	**1.0**	**1.0**	**1.0**
var	**1.0**	**1.0**	**1.0**	**1.0**
s1	**1.0**	**1.0**	**1.0**	**1.0**
var_n1	**1.0**	**1.0**	**1.0**	**1.0**
s_n1	-0.1	-0.1	-0.1	-0.1
cv	0.8	0.9	0.8	0.9
vr4	**1.0**	**1.0**	**1.0**	**1.0**
vr10	**1.0**	**1.0**	**1.0**	**1.0**
vr100	0.0	0.0	0.0	0.0
ap	0.0	0.0	0.0	0.0

Para evitar problemas en análisis futuros, se seleccionó solo una de las variables redundantes. Esta es la estrategia que se utilizó en todos los casos de correlación perfecta. Concluido el procedimiento las variables seleccionadas fueron: *p*, *b*, *t*, *f*, *d*, *ma*, *med*, *var*, *s*, *cv*, *vr*4, *vr*10, *vr*100, *ap*, *amod*, *aboy*, *curtosis* y *e*.

La Tabla 5.8 y la Tabla 5.9presentan la matriz de correlación para las variables del comportamiento y desempeño del algoritmo. Las matrices no presentan valores muy altos de correlación, lo cual es evidencia de no colinealidad. Por esta razón, todas las variables son utilizadas en análisis posteriores.

Tabla 5.8 Matriz de correlación de variables que miden el comportamiento del algoritmo

	m_b	*v_b*	*cl_s*	*v_cls*	*Ro*	*delta*	*Ho*	*Mo*	*e_opt*	*h0*	*engaño*
m_b	1.000	.071	.892	.050	.668	.425	.382	.429	.694	.659	.241
V_b	.071	1.000	-.087	.340	.281	.091	.090	.256	-.050	.175	.241
cl_s	.892	-.087	1.000	.023	.545	.300	.332	.456	.641	.659	.195
V_cls	.050	.340	.023	1.000	.075	-.216	-.109	.239	-.219	.039	.238
ro	.668	.281	.545	.075	1.000	.475	.752	.608	.552	.863	.213
delta	.425	.091	.300	-.216	.475	1.000	.368	.108	.884	.410	.168
Ho	.382	.090	.332	-.109	.752	.368	1.000	.403	.385	.802	.031
Mo	.429	.256	.456	.239	.608	.108	.403	1.000	.174	.643	.370
E_opt	.694	-.050	.641	-.219	.552	.884	.385	.174	1.000	.534	.167
h0	.659	.175	.659	.039	.863	.410	.802	.643	.534	1.000	.215
sk	.241	.241	.195	.238	.213	.168	.031	.370	.167	.215	1.000

Tabla 5.9 Matriz de Correlación de variables que miden el desempeño del algoritmo

	Pasos	*rcal*	*fit_error*	*m_f*	*p_f*	*m_fs*	*v_fs*
pasos	1.000	-.184	-.263	-.206	-.241	-.004	.605
rcal	-.184	1.000	.902	-.434	.318	-.854	-.409
fit_error	-.263	.902	1.000	-.575	.460	-.877	-.483
m_f	-.206	-.434	-.575	1.000	-.454	.685	.128
p_f	-.241	.318	.460	-.454	1.000	-.443	-.609
m_fs	-.004	-.854	-.877	.685	-.443	1.000	.329
v_fs	.605	-.409	-.483	.128	-.609	.329	1.000

Análisis de factores

En este paso el objetivo fue identificar un subconjunto de nuevas variables que presentara las mismas características que las variables originales. Se aplicaron algunas estrategias sugeridas por Johnson en [Johnson00] para el análisis de factores. El uso complementario de estas estrategias permitió obtener un patrón uniforme en la generación de nuevas variables para las versiones analizadas. Se consideraron los siguientes criterios en la aplicación del análisis de factores:

a. Extracción de factores usando el método Componentes Principales generado a partir del análisis de la matriz de correlación

b. Rotación de la solución usando el método Varimax
c. Estimación de las calificaciones de los factores utilizando regresión lineal simple
d. Los criterios seleccionados para determinar el número de factores a extraer fueron los siguientes, aplicados en el orden mostrado.
    1. Criterio MinEigen
    2. Análisis de la gráfica Scree
    3. Análisis de las estimaciones de las correlaciones reproducidas

De manera general, el procedimiento consistió en cuatro pasos: selección del número de factores a extraer, análisis de los valores de la matriz de componentes rotada, revisión de agrupaciones en la gráfica de las primeras tres componentes e interpretación de los factores. A continuación se muestra el resultado del análisis de factores para las variables propuestas.

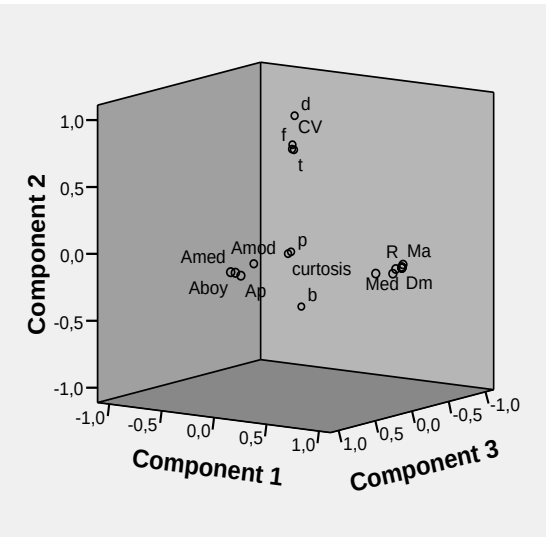
Análisis de factores de variables que miden la complejidad del problema

Se aplicó el análisis a las variables descritas en la Tabla 5.4. La Figura 5.1 presenta la matriz de componentes obtenida al extraer cuatro factores. Las variables que presentan mayor contribución en cada componente se presentan sombreadas y la gráfica muestra las agrupaciones obtenidas en las tres primeras componentes.
Analizando esta información y la descripción de las variables se dio la siguiente interpretación a las componentes asociadas a cada factor:

Componente 1. Las variables que presentan una mayor carga en la matriz rotada para esta componente son: *r*, *s*, *dm*, *ma*, *med*, *e*, *var*, *vr*10, *vr*100 y *vr*4. Estas variables son principalmente medidas de la tendencia central y dispersión de los pesos, por esta razón a este factor se le dio el nombre de *centralidad*.

Componente 2. En esta componente las variables que más influyen son: *d*, *cv*, *t* y *f*. Estas variables miden la variabilidad de los pesos respecto al contenedor, la variación y los factores existentes entre pesos, por lo que fue llamada variabilidad del problema: *variabilidad_p*.

	Component			
	1	2	3	4
R	,997	-,050	-,023	,005
s	,997	-,051	-,021	,005
Dm	,997	-,051	-,021	,005
Ma	,995	-,058	-,027	,010
Med	,994	-,058	-,031	,010
E	,994	-,057	-,028	-,048
Var	,994	-,034	-,044	,011
vr100	,988	-,054	,046	,018
VR10	,983	-,085	,079	-,056
VR4	,884	-,082	,170	-,058
d	-,080	,962	-,101	,046
CV	-,086	,747	-,081	-,071
t	-,040	,720	-,036	,136
f	-,080	,717	-,066	-,029
Amed	,041	-,045	,943	-,001
Aboy	,048	-,056	,889	-,022
Ap	,029	-,094	,788	,051
Amod	-,082	-,065	,452	,287
p	-,025	-,032	,026	,968
curtosis	-,021	-,038	,073	,966
b	-,047	-,468	-,146	-,669



Componente	Nombre del factor
1	*centralidad*
2	*variabilidad_p*
3	*asimetría*
4	*tamaño*

Figura 5.1 Análisis de factores para variables que representan la complejidad del problema

Componente 3. Para esta componente las variables que contribuyen con más carga en la matriz rotada son: *amed*, *aboy*, *ap* y *amod*. Estas variables miden la asimetría de la distribución de los pesos, por lo que este factor recibió el nombre de *asimetría*.

Componente 4. Para esta componente las variables que contribuyen con más carga en la matriz rotada son: *p*, *curtosis* y *b*. La variable *p* es una normalización del tamaño del problema, la *curtosis* indica la amplitud del aplanamiento de la distribución de pesos y *b* indica la proporción de la suma de objetos que es posible asignar al contenedor. Estas variables están asociadas al tamaño del problema, por este motivo el factor fue llamado *tamaño*.

Análisis de factores para variables relacionadas con el comportamiento del algoritmo

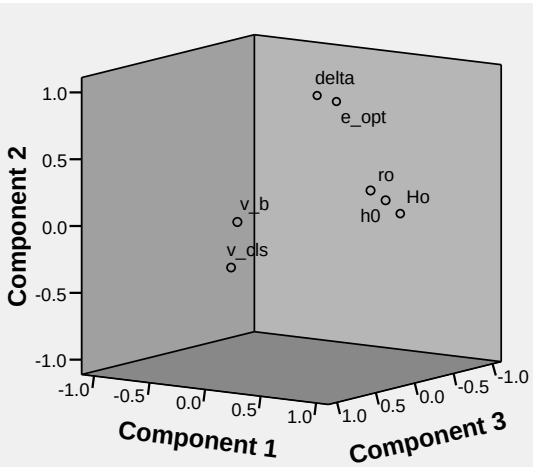
Se aplicó el análisis a las variables descritas en la Tabla 5.5. La Figura 5.2 presenta los resultados obtenidos. La interpretación que se dio a las componentes asociadas a cada factor es la siguiente:

Componente 1. Esta componente esta formada principalmente por las variables *Ho*, *h0*, *ro*, que son métricas derivadas del análisis de la superficie de aptitudes y miden la rugosidad de la trayectoria descrita por las soluciones del algoritmo. Es por esto que a este factor se le dio el nombre de *rugosidad*.

Componente 2. Las variables que presentan mayor contribución en esta componente son *delta*, *e_opt*, estas son también métricas del análisis de la superficie de aptitudes y miden la modalidad de la trayectoria descrita por las soluciones, por lo que el factor que las contiene fue llamado *modalidad*.

Componente 3. En esta componente las aportaciones principales corresponden a las variables *v_opt* y *v_cls*, que indican la varianza del valor de aptitud de las mejores soluciones y la varianza del valor de aptitud de los elementos en la lista de candidatos. Estas variables están relacionadas porque en el algoritmo Búsqueda Tabú, la lista de candidatos contiene las mejores soluciones que se generaron en la vecindad. El factor fue llamado *variabilidad_f*, porque representa la variabilidad de las aptitudes (fitness) que son consideradas óptimas.

	Component		
	1	2	3
Ho	.924	.126	-.081
h0	.915	.251	.095
ro	.859	.335	.209
delta	.217	.947	-.022
e_opt	.321	.899	-.121
v_b	.101	.114	.840
v_cls	.006	-.245	.784



Componente	Nombre
1	*rugosidad*
2	*modalidad*
3	*variabilidad_f*

Figura 5.2 Análisis de factores de variables relacionadas con el comportamiento del algoritmo

Análisis de factores para variables relacionadas con el desempeño del algoritmo

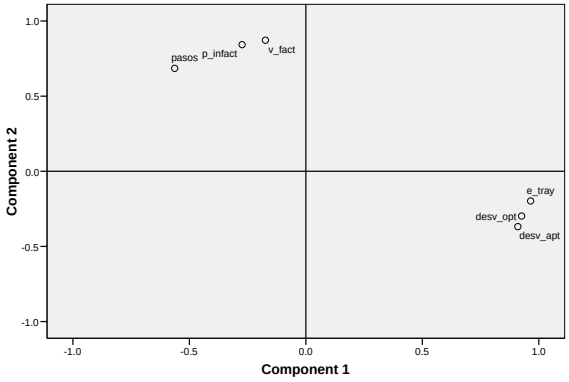
Se aplicó el análisis a las variables descritas en la Tabla 5.6 el método antes descrito. La Figura 5.3 presenta los resultados obtenidos. La interpretación que se dio a los factores encontrados es la siguiente:

Componente 1. Las variables que presentan mayor carga en esta componente son: *e_tray*, *desv_opt* y *desv_apt.* Estas variables representan el error obtenido durante la trayectoria, y las desviaciones de la mejor solución respecto al valor de aptitud óptimo por lo que el factor fue llamado *error*.

Componente 2. Esta formado principalmente por las variables *v_fact*, *p_fact*, y *pasos*. Estas variables miden la varianza de las soluciones factibles que se generaron en la trayectoria, el porcentaje de soluciones factibles encontradas y las evaluaciones de la

función de aptitud que realizo el algoritmo durante su ejecución. Se puede notar que éstas variables indican el esfuerzo computacional realizado por el algoritmo, por lo que el factor se denomino *esfuerzo*.

	Component	
	1	2
e_tray	.965	-.198
desv_opt	.926	-.298
desv_apt	.910	-.369
v_fact	-.173	.872
p_infact	-.274	.842
pasos	-.563	.685



Componente	Nombre
1	*error*
2	*Esfuerzo*

Figura 5.3 Análisis de factores para variables relacionadas con el desempeño del algoritmo

Indicadores del proceso algorítmico

Como se puede observar, aplicando el análisis de factores se obtuvo una reducción de la cantidad de variables utilizadas para el análisis. El patrón de agrupación mostrado se mantuvo para todas las versiones analizadas (ver anexo D). Encontrar esta clase de patrones ayuda en la comparación de los modelos causales.

Algunas de las variables propuestas no fueron incluidas en este proceso, porque durante el análisis se encontró que no tenían relación común con las demás. Estas variables y los factores generados fueron utilizados como indicadores de la complejidad del problema, comportamiento y desempeño del algoritmo para la creación del modelo causal. La Tabla 5.10 presenta el listado de los indicadores obtenidos una vez finalizado el proceso.

Tabla 5.10 Indicadores del proceso algorítmico

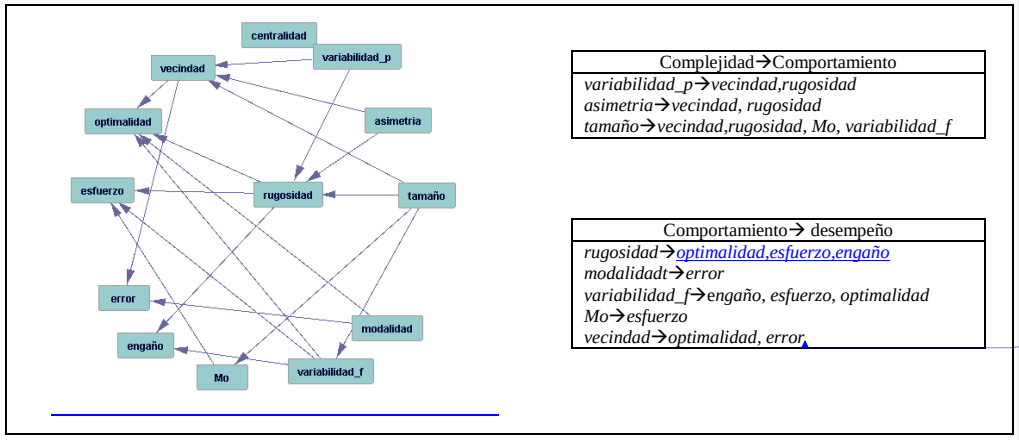
Tipo de Indicador	Indicadores
Complejidad del Problema	*centralidad, variabilidad_p, asimetría, tamaño*
Comportamiento del Algoritmo	*rugosidad, modalidad, variabilidad_f, vecindad*
Desempeño del Algoritmo	*error, esfuerzo, decepcion, optimalidad*

Con formato: Centrado
Con formato: Justificado
Con formato: Español (alfab. internacional)
Con formato: Justificado
Con formato: Justificado

5.3.3 CREACIÓN DEL GRAFO CAUSAL

Se crearon los modelos según la metodología propuesta, incluyendo el conocimiento previo que se tiene de las posibles relaciones entre complejidad del problema, comportamiento y desempeño del algoritmo:

1. Indicadores de comportamiento←Indicadores de complejidad
2. Indicadores de desempeño← Indicadores de comportamiento



Con formato: Inglés (Estados Unidos)

Figura 5.4 Grafo causal para indicadores de complejidad, comportamiento y desempeño de la versión v1 del algoritmo Búsqueda Tabú

Se utilizó el algoritmo PC disponible en el programa TETRAD para crear el grafo causal que representa las relaciones causales para los indicadores de complejidad del problema, comportamiento y desempeño. La Figura 5.4 muestra el grafo y el listado de las relaciones encontradas.

5.3.4 ESTIMACIÓN DEL GRAFO CAUSAL

Se estimaron las relaciones causales utilizando el estimador de ecuaciones estructurales disponible en TETRAD. Este produce una estimación completa de los parámetros del modelo utilizando el método de máxima verosimilitud [Johnson02].

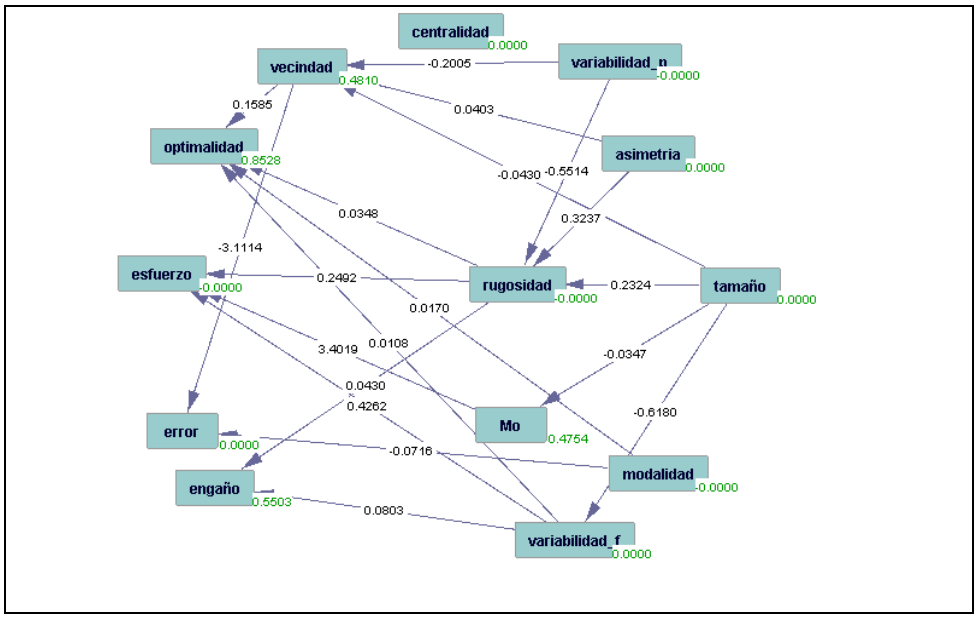


Figura 5.5 Estimación del grafo causal

Para las variables causa el valor en la arista corresponde a la contribución de éstas sobre la variable efecto, mientras que el coeficiente de la variable efecto representa una medida del error en que se incurre si no se hubieran incluido un número adecuado de variables causa.

La Figura 5.5 presenta el modelo de ecuaciones estructurales obtenido para el grafo causal obtenido, donde cada ecuación representa una relación causal. En la siguiente sección se proporciona la interpretación de este modelo.

5.3.5 VALIDACIÓN DEL GRAFO CAUSAL

Para validar el grafo causal generado, se realizó una prueba de hipótesis con el estadístico χ^2 , bajo los supuestos descritos en la sección 4.3.3. Donde:

H_0: La matriz de covarianzas de la población es igual a la matriz de covarianza estimada para las variables medidas.

H_1: La matriz de covarianzas de la población no es igual a la matriz de covarianza estimada para las variables medidas.

El resultado de la prueba, utilizando 59 grados de libertad y un nivel de significación $\alpha = 0.05$ es: χ^2 =2107.6178, con un valor P = 0.0000. Lo que indica que la hipótesis nula se acepta, y entonces el modelo estimado si representa la estructura causal subyacente a los datos a partir de la cual fue generado.

5.4 INTERPRETACIÓN DE LAS RELACIONES CAUSALES

Para interpretar las relaciones causales, se analizó el modelo de ecuaciones estructurales mostrado en la Figura 5.5 y se tomo en cuenta el conocimiento previo que se tiene del problema. A continuación se interpretan las ecuaciones de regresión asociadas a las relaciones causales para la versión 1 del algoritmo Búsqueda Tabú.

vecindad = 0.4810 – 0.2005 *variablidad_p* + 0.0403 *asimetria* – 0.0430 *tamaño.* El tamaño de la lista de candidatos puede expresarse en función de la variabilidad de los valores de aptitud, asimetría y tamaño del problema. Si la variabilidad y el tamaño de los pesos asociados a los objetos son grandes, el tamaño de la vecindad disminuye.

optimalidad = 0.8528 + 0.0348 *rugosidad* + 0.0170 *modalidad* + 0.0108 *variabilidad_f* + 0.1585 *vecindad.* La optimalidad del algoritmo, mide el grado de convergencia hacia un valor de aptitud óptimo. La expresión asociada indica que el algoritmo tardara más en

detener la exploración si los elementos que definen la navegación, como la rugosidad, modalidad, variabilidad y el tamaño de la vecindad explorada, siguen en valores altos.

esfuerzo = 0.2492 *rugosidad* + 3.4019 *Mo* + 0.4262 *variabilidad_f*. El esfuerzo que realiza el algoritmo se incrementa si el espacio en el que navega es muy rugoso, con múltiples óptimos locales y si las soluciones que encuentra presentan alta variabilidad.

error = – 3 .1114 *vecindad* – 0.0716 *modalidad.* El error que obtiene el algoritmo depende mucho del grado de exploración que realiza y de la presencia de óptimos locales que confunden al algoritmo durante la búsqueda de soluciones. El error disminuye si la vecindad explorada es mayor y existen pocos óptimos locales, permitiendo así un mejor desempeño del algoritmo.

engaño = 0.5503 + 0.0430 *rugosidad* + 0.0803 *variabilidad_f*. El indicador de engaño es una medida que permite saber que tan complicado será resolver el problema. Dado que la rugosidad del espacio de solución y la variabilidad de los valores de aptitud influyen en la trayectoria que sigue el algoritmo, es de esperarse que un valor alto de estos indicadores genere un coeficiente de engaño alto.

variabilidad_f = – 0.6180 *tamaño.* El indicador del tamaño del problema es causa de la variabilidad de la función de aptitud. A mayor tamaño del problema menos variabilidad, esto es porque la función de aptitud permite diferenciar entre cambios en la distribución de los objetos. Si son muchos los objetos a distribuir, un intercambio entre ellos no generará cambios drásticos en la función de aptitud.

Mo = 0.4754 – 0.0347 *tamaño*. Esta relación es similar a la anterior *Mo* mide los cambios en la trayectoria del algoritmo, si hay muchos objetos en el problema disminuye la variabilidad de la función de aptitud y por consecuencia los cambios en la trayectoria serán menores.

rugosidad = 0.2324 *tamaño* + 0.3237 *asimetria* – 0.5514 *varibilidad_p*. La rugosidad mide la correlación que tienen los valores de aptitud de las soluciones que se generan durante la ejecución del algoritmo estos, como lo indican las relaciones anteriores, están afectados por el tamaño, asimetría, y variabilidad de los pesos de los objetos a distribuir.

5.5 EVALUACIÓN DE GRAFOS CAUSALES

El modelo de ecuaciones estructurales obtenido a partir de la especificación y estimación de la estructura causal funciona como un mecanismo que permite actualizaciones entre valores de variables sin perder la consistencia del modelo.

Este modelo puede ser utilizado como un mecanismo de predicción, haciendo posible intervenir las variables del modelo cambiando los valores de las variables y analizando los cambios observados en los coeficientes de las ecuaciones estructurales.

5.6 EXPLICACIÓN DEL DESEMPEÑO DEL ALGORITMO BÚSQUEDA TABU

Para el explicar por qué una versión se comportó mejor que otra en la solución de los casos, se analizaron comparaciones entre cambios de estrategias para obtener la solución inicial, el tamaño de la lista tabú y la selección de los operadores utilizados en la búsqueda. Se partió de una configuración inicial y se analizó si el cambio de estrategia afecta el desempeño del algoritmo y por consecuencia su estructura causal.

Siguiendo la metodología propuesta, se realizaron comparaciones entre las versiones generadas, para identificar aquellas que mostraran diferencias en su desempeño. La comparación se hizo entre versiones que solo varían un factor a la vez y aplicando los siguientes criterios:

1. Desviación del óptimo teórico: la versión mejor es aquella que tiene menor desviación al óptimo

2. Valor de aptitud obtenido. Si ambas versiones obtienen el mismo valor de desviación al óptimo, elegir como mejor a aquella que tenga un mejor valor de aptitud
3. Número de evaluaciones de la función objetivo. Si ambas versiones obtienen el mismo valor de desviación al óptimo, y una tiene menor valor de aptitud y menor número de evaluaciones, elegirla como mejor.

A continuación se presentan dos experimentos realizados para evaluar si el algoritmo de un algoritmo se ve afectado por cambios en su configuración.

5.6.1 EXPERIMENTO 1. COMPARACIÓN DE MODELOS CAUSALES PARA CAMBIOS EN LA SOLUCIÓN INICIAL

Se compararon los modelos causales para evaluar el cambio en la configuración de algoritmos cuando se utiliza una solución inicial aleatoria ó heurística. Para verificar la existencia de relación entre la forma en como se genera la solución inicial y el desempeño que obtiene el algoritmo se analizaron los modelos causales asociados a las comparaciones de la Tabla 5.11.

Tabla 5.11 Comparaciones usando como criterio la solución inicial

Comparación	Criterios fijos				Criterio de comparación		Versión campeona
	Parámetros de control		Operador de Búsqueda		Solución inicial		
	Tamaño de lista tabú Fija	Tamaño de lista tabú auto-configurada	Aleatorio	Único	Aleatoria	Heurística	
V1 vs V3	✓	--	✓		V1	V3	V1
V2 vs V4	✓	--		✓	V2	V4	V4
V5 vs V8	--	✓	✓		V5	V8	V5
V6 vs V7	--	✓		✓	V6	V7	V7

En las siguientes secciones se presenta el análisis de las estructuras causales asociadas a cada comparación. Tomando como referencia los principales elementos que afectan el comportamiento del algoritmo y su desempeño.

Comparación 1

El criterio para esta comparación es que la versión 1 utiliza una solución inicial generada de manera aleatoria, mientras que la versión 3 la genera a través de una heurística. En ambos casos la búsqueda se realiza a través de cuatro operadores seleccionados de forma aleatoria. En esta comparación la versión 1 mostró mejor desempeño. La Tabla 5.12 muestra las relaciones causales de error y esfuerzo de las versiones 1 y 3.

Tabla 5.12 Reglas causales para error y esfuerzo en comparación de versiones 1 y 3

Versión	Error	Esfuerzo
v1	*error*=-3.114*vecindad*-0.0716*modalidad*	*esfuerzo*=0.2492*rugosidad*+3.4019*Mo*+0.4262*variabilidad_f*
v3	*error* = -2.6970*vecindad*-0.1503*rugosidad*	*esfuerzo*= 4.3678*Mo*+0.4298*variabilidad_f* +0.3857*modalidad*

Ambas versiones generan una vecindad grande debido a que aplican múltiples operadores, sin embargo el punto a partir del cual inician la búsqueda determina el grado de error resultante.

Para las instancias analizadas, la versión 3 inicia en un punto alto en la superficie del problema y su error se incrementa debido a que partir de este punto las vecindades son más pequeñas porque las soluciones se encuentran mas correlacionadas.

La versión 1 tiene un mejor desempeño porque inicia en un punto mas bajo en la superficie del problema y le afecta principalmente la existencia de múltiples óptimos locales (modalidad).

Comparación 2

En esta comparación, la versión 2 utiliza una solución inicial generada de manera aleatoria, mientras que la versión 4 la genera a través de una heurística. En ambos casos para la búsqueda se utiliza un solo operador que consiste en un intercambio 1-1. La versión 4 mostró mejor desempeño. La Tabla 5.13 muestra las reglas asociadas al error y desempeño de cada versión comparada.

Tabla 5.13 Reglas causales para error y esfuerzo en comparación de versiones 2 y 4

Versión	Error	Esfuerzo
v2	*error* = -3.4532*vecindad*-0.2449*rugosidad*	*esfuerzo* = -3.5949*Mo*+ 0.2951*variabilidad_f* +0.5748*modalidad*
v4	*error* = -1.9955*vecindad*	*esfuerzo*=-0.8973Mo+0.5478*modalidad* +3.1343*vecindad*

La versión 2 inicia en un punto bajo en la superficie del problema y trata de mejorar su valor de aptitud por medio de un solo operador, pero como este es limitado no obtiene mejores resultados. La versión 4 inicia un punto más alto y utilizando un solo operador logra superar a la versión 2.

Como se puede ver en la relación causal de *error* para la versión 4 éste causado por el tamaño de la vecindad, es decir, a mayor tamaño de vecindad menor error. Esto se debe a que en un punto alto, al abarcar un área más grande durante la exploración, se logra encontrar mayores diferencias entre los valores de aptitud. El esfuerzo realizado por el algoritmo depende de los cambios en la trayectoria, del número de óptimos encontrados y del tamaño de vecindad que se explore.

Comparación *3*

En esta comparación, permanecen fijos el tamaño de lista tabú, que se obtiene de acuerdo al problema, y la selección del operador que es aleatoria. La versión 5 utiliza una solución inicial generada de manera aleatoria, y la versión 8 una construcción

heurística. La versión que mostró mejor desempeño fue la versión 5. La Tabla 5.14 muestra las reglas asociadas al error y desempeño de cada versión comparada.

Tabla 5.14 Reglas causales para error y esfuerzo en comparación entre versiones 5 y 8

Versión	Error	Esfuerzo
v5	-2.7697*vecindad*-0.3218*rugosidad*-0.3196*variabilidad*	*esfuerzo* = – 0.6936 *modalidad* + 0.2805 *variabilidad_f*
v8	-3.1121*vecindad*-2.1378Mo-0.1198*rugosidad*	*engaño* = 0.5992 + 0.0193 *variabilidad_f*

La diferencia en la regla causal que genera el error entre las versiones 5 y 8 es que en la versión 5 el error disminuye al incrementar la variabilidad de los valores de aptitud y en la versión 8 el error disminuye al encontrar muchas diferencias en la trayectoria.

Debido a que la versión 8 inicia en un punto alto de la superficie, las aptitudes son similares y no existen muchas diferencias en la trayectoria generada. Mientras que la versión 5 al iniciar en un punto mas bajo, encuentra más variabilidad en los valores de aptitud que genera.

Comparación 4

En esta comparación el tamaño de la lista tabú se calcula en función del tamaño del problema y se utiliza solo un operador de búsqueda. La versión 6 utiliza una solución inicial generada de manera aleatoria, y la versión 7 una construcción heurística.

Tabla 5.15 Reglas causales para error y esfuerzo en comparación entre versiones 6 y 7

Versión	Error	Esfuerzo
v6	*error* =-3.4582+2.44*rugosidad*	*esfuerzo* = 0.0008 + 0.5120 *rugosidad* + 0.6409 *modalidad* + 0.3079 *variabilidad_f*
v7	*error*=-2.2755*vecindad*+1.8230*Mo*	*esfuerzo* = 0.2231 *rugosidad* + 0.1626 *variabilidad_ f* + 0.4092 *modalidad* + 1.0514*engaño*

La Tabla 5.15 muestra las reglas asociadas al error y desempeño de cada versión comparada. La versión que obtuvo mejor desempeño fue la 6, esto debido a que su error disminuye cuando las soluciones están menos correlacionadas, y esto sucedió al iniciar en un punto bajo de la trayectoria. Mientras que la versión 7 empezó en un punto alto, en donde se pueden presentarse muchos óptimos locales, no fue capaz de escapar de ellos utilizando un solo operador.

Conclusiones

En las comparaciones mostradas se puede observar que, no importando si la lista tabú es fija o adaptada al tamaño del problema, la mejor opción para lograr un buen desempeño del algoritmo partiendo de una solución heurística es utilizar un solo operador para mejorar la búsqueda (v4 y v8). Mientras que partiendo de una solución aleatoria, es recomendable utilizar múltiples operadores (v1 y v5).

5.6.2 EXPERIMENTO 2. COMPARACIÓN DE MODELOS CAUSALES PARA CAMBIOS EN LA SELECCIÓN DE LOS OPERADORES DE BÚSQUEDA

El objetivo de este experimento es la comparación de estructuras causales para evaluar el cambio en la configuración de algoritmos cuando la búsqueda de soluciones se hace con múltiples operadores seleccionados de forma aleatoria ó un operador fijo.

Para verificar la existencia de relación entre la forma en que se aplican los operadores de búsqueda y el desempeño que obtiene el algoritmo se analizaron los modelos causales asociados a las comparaciones de la Tabla 5.16.

Como se puede observar, presentan mejor desempeño las versiones que utilizan una búsqueda con varios operadores seleccionados al azar, sin importar la generación de la solución inicial. Esta situación se hizo evidente en el experimento 1 cuando las

versiones 1 y 5, que utilizan esta misma estrategia para la búsqueda mostraron mejor desempeño.

Tabla 5.16 Comparaciones usando como criterio el operador de búsqueda

Comparación		Criterios fijos				Criterio de comparación		Versión campeona
		Parámetros de control		Solución inicial		Operador de Búsqueda		
		Tamaño de lista tabú Fija	Tamaño de lista tabú auto configurada	Aleatoria	Heurística	Aleatorio	Único	
1	V1 vs V2	✓	--	✓		V1	V2	V1
2	V3 vs V4	✓	--		✓	V3	V4	V3
3	V5 vs V6	--	✓	✓		V5	V6	V5
4	V7 vs V8	--	✓		✓	V7	V8	V8

A continuación se presenta el análisis de estas comparaciones:

Comparación 1

En esta comparación se mantiene fijos el tamaño de la lista tabú y la solución inicial aleatoria. El cambio en la configuración es: v1 operadores al azar, v2 utilizó solo un operador. La Tabla 5.17 presenta las reglas causales asociadas al error y desempeño de las versiones analizadas.

Tabla 5.17 Reglas causales para error y esfuerzo en comparación entre versiones 1 y 2

Versión	Error	Esfuerzo
v1	$error$=-3.114$vecindad$-0.0716$modalidad$	$esfuerzo$ =0.2492$rugosidad$ +3.4019Mo+ 0.4262$variabilidad_f$
v2	$error$ = -3.4532$vecindad$-0.2449$rugosidad$	$esfuerzo$= -3.5949Mo+0.2951$variabilidad_f$+0.5748$modalidad$

Al utilizar ambas versiones una solución inicial aleatoria, es posible que el recorrido inicie en un punto bajo de la superficie de aptitud. La versión 1 difícilmente mejora su valor de aptitud porque solo tiene un operador con el cual realizar la búsqueda

y esto le da pocas opciones de exploración. La versión 1 por otra inicia también con un valor de aptitud bajo, pero el tamaño de la vecindad que explora es mayor, por lo que tiene mas posibilidades de encontrar el óptimo global.

Las reglas causales asociadas al error y esfuerzo de la versión 1 indican que el desempeño es afectado principalmente por la capacidad de exploración del algoritmo, ya que esta última es determinada en un alto grado por la existencia de óptimos locales y la variabilidad de las aptitudes de las soluciones encontradas.

Comparación 2

En esta comparación, se mantienen fijos el tamaño de la lista tabú y la estrategia para seleccionar la solución inicial, que es heurística. Las versiones 3 y 4 son comparadas por la selección de los operadores de búsqueda; múltiples operadores seleccionados de forma aleatoria y un operador fijo respectivamente. La versión 3 mostró mejor desempeño. Las relaciones causales para ambas versiones se muestran en la Tabla 5.18.

Tabla 5.18 Reglas causales para error y esfuerzo en comparación entre versiones 3 y 4

Versión	Error	Esfuerzo
v3	*error* = -2.6970*vecindad*-0.1503*rugosidad*	*esfuerzo*= 4.3678Mo+0.4298*variabilidad_f* 0.3857*modalidad*
v4	*error* = -1.9955*vecindad*	*esfuerzo* = -0.8973*Mo* +0.5478*modalidad* +3.1343*vecindad*

La versión v3 mostró mejor desempeño que la versión v4 y presenta el mismo patrón que la comparación anterior, iniciando desde un punto mas alto se tiene mayor posibilidad de mejorar el valor de aptitud utilizando varios operadores.

El esfuerzo que el algoritmo realiza en el recorrido a partir de un valor alto de aptitud se ve influido por la cantidad de cambios se generan en la trayectoria, el numero de óptimos locales en el recorrido y el tamaño de la vecindad. Conforme el tamaño de la vecindad evaluada aumenta, el esfuerzo también.

Comparación 3

En esta comparación, se mantienen fijos el tamaño de la lista tabú (calculado en base al problema) y la estrategia para seleccionar la solución inicial, que es aleatoria. Las versiones 5 y 7 son comparadas por la selección de los operadores de búsqueda; múltiples operadores seleccionados de forma aleatoria y un operador fijo respectivamente.

La versión que obtuvo mejor desempeño fue la versión 5. La Tabla 5.19 presenta las relaciones causales encontradas para los indicadores de error y desempeño de las versiones comparadas.

Tabla 5.19. Reglas causales para error y esfuerzo en comparación entre versiones 5 y 6

Versión	Error	Esfuerzo
v5	*error* = – 2.7697 *vecindad* – 0.3218 *rugosidad* – 0.3196 *variabilidad_*	*esfuerzo* = – 0.6936 *modalidad* + 0.2805 *variabilidad_f*
v6	*error* = 0.0012 – 3.4582 *vecindad* + 0.2446 *rugosidad*	*esfuerzo* = 0.0008 + 0.5120 *rugosidad* + 0.6409 *modalidad* + 0.3079 *variabilidad_f*

Para esta versión el error disminuye cuando se aumenta el tamaño de la vecindad, y las soluciones están poco correlacionadas entre si. Por otra parte, la correlación de las soluciones que encuentre, el número de óptimos locales y la variabilidad de los valores de aptitud definen el esfuerzo realizado. Esto principalmente porque al utilizar una lista tabú más grande, obliga al algoritmo a realizar una mayor exploración para encontrar soluciones candidatas,

Comparación 4

En esta comparación, se mantienen fijos el tamaño de la lista tabú (calculado en base al problema) y la estrategia para seleccionar la solución inicial, que es heurística. Las versiones 8 y 7 son comparadas por la selección de los operadores de búsqueda; múltiples operadores seleccionados de forma aleatoria y un operador fijo respectivamente.

Tabla 5.20 Reglas causales para error y esfuerzo en comparación entre versiones 7 y 8

Versión	Error	Esfuerzo
v7	*error* = - 2.2755 *vecindad* + 1.8230 *Mo*	*esfuerzo* = 2.2224 *vecindad* + 0.3380 *variabilidad_f* + 0.4529 *modalidad*
v8	*error* = – 3.1121 *vecindad* – 0.1198 *rugosidad* – 2.1378 *Mo*	*esfuerzo* = 0.2231 *rugosidad* + 0.1626 *variabilidad_ f* + 0.4092 *modalidad* + 1.0514 *engaño*

En esta comparación se desempeño mejor la versión 8, como se observa en la estructura causal, ver Tabla 5.20, en ambas el error esta definido por la vecindad y los cambios que existen en la trayectoria. Sin embargo, en la versión 8 a mayor rugosidad menor error, ya que la combinación de estrategias que la componen permite que tome ventaja de los picos y valles presentes en la superficie de búsqueda.

Como la versión 8 navega a través de espacios rugosos, con múltiples óptimos locales realiza un mayor esfuerzo. La aportación de la variabilidad de la función de aptitud en el esfuerzo tiene que ver con la condición de estabilidad del algoritmo, ya al obtener valores muy variados tardara más en converger a una solución estable.

Conclusión

Como puede observarse en las comparaciones, la mejor estrategia para resolver el problema, es utilizar una solución inicial aleatoria, y realizar la búsqueda con varios operadores seleccionados al azar. También se concluye de acuerdo al análisis de las estructuras causales, que los elementos que más influyen en el desempeño del algoritmo, cuando este es adecuado para resolver el problema son: tamaño de la vecindad, correlación de las soluciones y variabilidad de la función objetivo.

Capítulo 6

CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las conclusiones de este trabajo, así como también sugerencias para el desarrollo de trabajos futuros

6.1 *CONCLUSIONES*

El trabajo de investigación presentado aborda el análisis del desempeño de algoritmos metaheurísticos cuando estos resuelven problemas de distribución de objetos. En particular se presenta el análisis de algoritmos metaheurísticos que resuelven el problema de Bin Packing.

La principal aportación de este trabajo es el desarrollo de una **metodología para la construcción sistemática de modelos causales** que representan las relaciones entre la complejidad del problema, el comportamiento del algoritmo y el desempeño obtenido por el mismo.

El modelado causal aplicado al análisis de algoritmos permite explicar con rigor estadístico, cómo la naturaleza del problema y la estructura de diseño de los algoritmos afectan su desempeño. Estudios a profundidad con este tipo de modelado posibilitan el establecimiento de bases teóricas, la evaluación de algoritmos y el diseño de algoritmos con mejor desempeño.

Se presentó el análisis detallado para un caso de estudio aplicando la metodología propuesta, y los resultados indican que es factible aplicar este procedimiento a otros algoritmos metaheurísticos que resuelven problemas de optimización combinatoria referente la distribución de objetos.

Las principales contribuciones de este trabajo, son las siguientes:

1. El desarrollo de un enfoque sistemático formal para el análisis experimental del desempeño de estrategias metaheurísticas. En el enfoque tradicional se presentan estadísticas de desempeño para evaluar algoritmos en torneos de competencia. Los trabajos más recientes constituyen modelos de predicción del desempeño. Esta tesis construye modelos causales que permiten estudios más profundos del desempeño algorítmico.

2. Se presenta la formulación de indicadores que describen al problema, el comportamiento del algoritmo y su desempeño.

3. Una estrategia para generar e interpretar modelos causales a partir de indicadores del proceso algorítmico para identificar relaciones entre problema, algoritmo y desempeño.

4. El modelo causal del desempeño del algoritmo Búsqueda Tabú.

5. Explicaciones formales del por qué algunas configuraciones de parámetros de control del algoritmo Búsqueda Tabú influyen de manera directa en el desempeño mostrado por el algoritmo.

6. Una estrategia para generar explicaciones formales de las relaciones existentes entre los elementos que afectan el desempeño del algoritmo.

6.2 *TRABAJO FUTURO*

Para dar continuidad al trabajo de investigación presentado se proponen los siguientes trabajos:

a. Formular nuevos indicadores de complejidad del problema, comportamiento y desempeño del algoritmo. Es deseable que estos indicadores complementen las propuestas de esta tesis o mejoren su capacidad de representación.

b. Utilizar el modelado causal como parte de un esquema de selección de algoritmos basado en predicción.

c. Utilizar el conocimiento generado del análisis causal del comportamiento del algoritmo analizado para la mejora de su desempeño, adaptando su estructura de diseño al problema que resuelven

d. Ampliar la metodología propuesta mejorando el diseño factorial de los experimentos necesarios para evaluar el impacto de la configuración de parámetros. En este trabajo se utilizó un diseño factorial completo con solo dos variaciones por parámetro. Si se desea incluir más variaciones de parámetros para mejorar el análisis, el esfuerzo computacional es prohibitivo. Es por esto que se propone utilizar un diseño factorial fraccional [Montgomery04] que permita identificar estadísticamente cuántas configuraciones analizar para obtener buenos resultados y un esfuerzo computacional razonable

e. Extender la metodología a otra clase de problemas, por ejemplo estudiar las relaciones entre propiedades de grafos y problemas de grafos.

Anexo A

DOCUMENTACIÓN DEL ALGORITMO TABU

En este anexo se presenta la documentación relacionada con la implementación de las versiones del algoritmo Búsqueda Tabú para resolver el problema de Bin Packing.

A.1 Función de aptitud para guiar la búsqueda de soluciones del problema Bin Packing

Para guiar al algoritmo a través de buenas soluciones se requiere evaluar la calidad de las soluciones. En el caso de Bin Packing, el número de contenedores requeridos *m* representa el valor de la solución obtenida. Frecuentemente se presentan muchas soluciones posibles con el mismo número de contenedores intercambiando los objetos que hay entre ellos.

Si las soluciones posibles tienen el mismo número de contendores, es más complicado para el algoritmo continuar la búsqueda de soluciones. Falkenauer propone la función de aptitud mostrada en la expresión 1 que supone una mejor discriminación cuando se presentan las situaciones descritas [Falkenauer96].

$$f(s) = \frac{\sum_{i=1}^{n} \left(F_i / C \right)^k}{N} \quad (1)$$

En la expresión N es el número de contenedores, F_i el total almacenado en el contenedor i, C es el contenido máximo de un contenedor y k es un parámetro fijo con valor 2.

A.2 Operadores para generar vecindades para el problema Bin Packing

En la solución del problema de Bin Packing, para generar soluciones vecinas a partir de una solución dada, se intercambian objetos entre todos los pares posibles de contenedores. En este trabajo se implementaron cuatro operadores de intercambio de objetos [Fleszar02, Loh06] que son:

a. Intercambio 1-0. Mover un objeto del contenedor α al contenedor β,
b. Intercambio 1-1. Intercambiar un objeto i del contenedor α con un objeto j contenedor β
c. Intercambio 1-2. Mover un objeto i del contenedor α con los objetos j y k del contenedor β
d. Intercambio 2-2. intercambiar el par de objetos i y j del contenedor α con los objetos j y l del contenedor β

Cuando estos operadores se utilizan para generar vecindades a partir de una solución inicial, el valor de aptitud se calcula de manera parcial, evaluando solo la contribución del intercambio aplicado y esto disminuye el tiempo de procesamiento.

A.3 Algoritmo Búsqueda Tabú básico

A continuación se presentan el pseudocódigo utilizado para la implementación de algoritmo Busqueda Tabu tomado como base(Figura A.1). Los parámetros de configuración que permiten generar las diferentes versiones (ver Tabla 1.1), y las

condiciones de terminación del algoritmo, que fueron las mismas para todas las versiones implementadas.

```
gBS: Mejor solución global
listSize: Tamaño de la lista tabú
candidates: soluciones candidatas de la vecindad
listSize 0.8*raiz(n)
Mientras no se alcance el criterio de terminación
      iBS: Mejor solución de la iteración
      iBS ← gBS
      n_neighbor ← 0.8*raiz(n)   // Cantidad de Vecinos
      current ? iBS
      Hacer
            Para I ← 0 hasta I = nNeighbor
                  Generar un movimiento de manera aleatoria
                  (almacenarlo en movement)
                  Si movement está en lista tabú
                        Ignorarlo y volver al inicio del ciclo
                  Fin Si
                  Si es posible generar vecino a partir de
                  movement
                        Ignorarlo y volver al inicio del ciclo
                        Si el vecino pasa el umbral de aceptación
                              Agregar neighbor a candidates
                        Fin Si
                  Fin Si
            Fin Para
            Si existen candidatos
                  Seleccionar un candidato de acuerdo a su
                  desempeño usando el método de ruleta
                  Almacenar en lista tabú el inverso del
                  movimiento que generó a alea
                  Si el desempeño de alea es mayor al de iBS
                        iBS ← current
                  Fin Si
                  no_neighbor ← 0
            Sino
                  no_neighbor ← no_neighbor + 1
            Fin Si
            Decrementar el umbral de aceptación de acuerdo a
            THRESHOLDING_DECREMENT
            Actualizar la lista tabu
      Mientras no_neighbor < NO_NEIGHBORHOOD
      Si iBS tiene mejor desempeño que gBS
            gBS ← iBS
      Fin Si
      Restaurar el umbral a su valor inicial
Fin mientras
```

Figura A.1 Pseudocódigo del Algoritmo Búsqueda Tabu

Tabla A.1 Parámetros de configuración:

MAX_ITERATIONS: Número máximo de iteraciones (4000)
THRESHOLDING: Umbral inicial de aceptación (0.10)
THRESHOLDING_DECREMENT: Factor de disminución del umbral (0.90)
NO_NEIGHBORHOOD: Número máximo de vecindades sin que se acepten vecinos (5)
MaxCoincidences. Número de veces que se permite una solución repetida (tamaño de lista tabú)
EQUALITY: Factor de igualdad (0.0001)

Condiciones de terminación del algoritmo:

a. Se ha repetido la misma solución maxCoincidences (tamaño de lista tabú) veces y con esto se produce un estancamiento del algoritmo)
b. Se alcanza el óptimo teórico (convergencia)
c. Se termina el máximo de iteraciones MAX_ITERATIONS (divergencia)

A.4 Diagrama de clase de la implementación del algoritmo Búsqueda Tabú

Para la implementación del algoritmo crearon las siguientes clases:

TabuList: es una estructura de memoria para almacenar los movimientos calificados como tabú. Esta clase almacena los movimientos y el tiempo máximo que debe estar un elemento en la lista esta determinado por el tamaño de la misma. El método *update* verifica el tiempo de tenencia de los movimientos tabú y saca de la lista aquellos elementos cuyo tiempo de tenencia ha terminado.(ver Figura A.2)

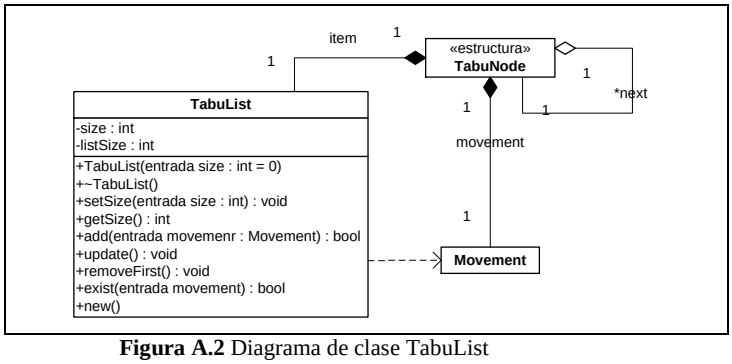


Figura A.2 Diagrama de clase TabuList

TabuSearch: realiza el proceso del algoritmo utilizando el método solve. (Figura A.3)

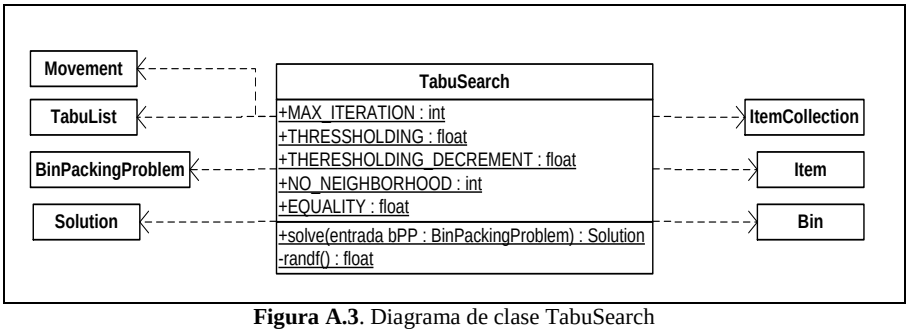


Figura A.3. Diagrama de clase TabuSearch

El diagrama de clases global para el algoritmo de búsqueda tabú es el mostrado en la Figura A.4.

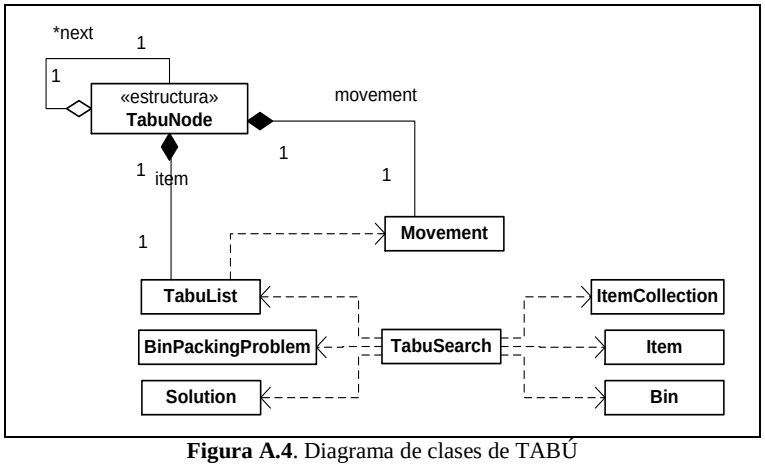


Figura A.4. Diagrama de clases de TABÚ

A5. Evaluación del desempeño del algoritmo

Para lograr que la evaluación de las implementaciones utilizadas fuera consistente, se unificaron los siguientes aspectos:

- La representación computacional de una solución del problema: las soluciones son listas enlazadas.

- Esquema de vecindad utilizado: cuatro operadores basados en intercambios entre objetos.
- Solución inicial aleatoria o heurística dependiendo de la configuración

Se evaluó el desempeño de la implementación base utilizando como referencia de comparación una implementación reportada en Cruz 2004 [Cruz04]. En la Figura 4 se presenta una gráfica de comparación de ambas implementaciones utilizando como referencia la calidad de la solución obtenida.

Como se puede observar, la versión que se utiliza en este trabajo (etiquetada como nuevo) presenta una mejoría notoria respecto a la reportada en [Cruz04] (etiquetada como anterior). La implementación presentada muestra una mejora del 80% sobre las instancias consideradas.

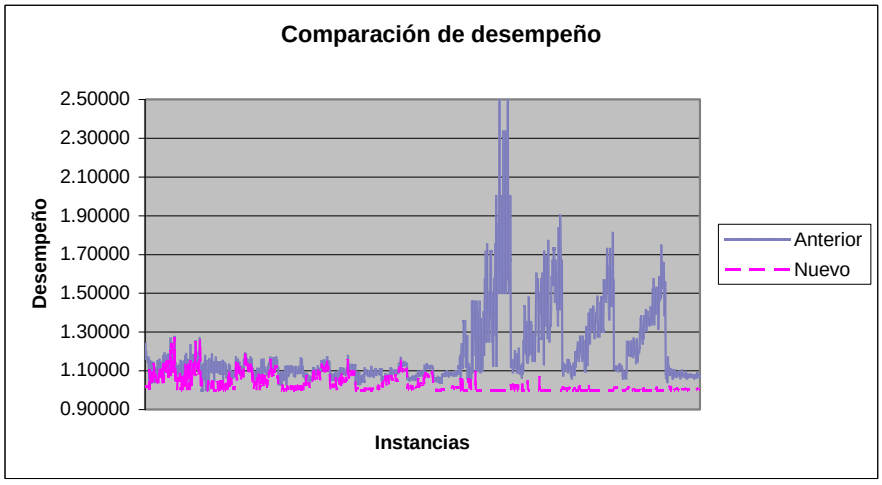


Figura A.5 Desempeño de versión anterior vs. versión nueva del algoritmo Búsqueda Tabu

Anexo B

EJEMPLO DE CÁLCULO DE MÉTRICAS DE ANÁLISIS DE LA SUPERFICIE DE APTITUDES

En este anexo se presenta un ejemplo del cálculo de las métricas descritas en la sección 2.5. Para desarrollarlo se utilizan los valores de aptitud, ‘estos fueron obtenidos de la trayectoria del algoritmo Tabú durante una ejecución. Una trayectoria son todas las soluciones que son generadas por el algoritmo hasta que alcanza su criterio de parada.

En la Tabla B.1 se presenta una muestra de 10 soluciones tomadas a una distancia 1 (generadas consecutivamente). Los valores de aptitud finales son los valores finales de una trayectoria, en la tabla se presentan los correspondientes a diez corridas del algoritmo.

Tabla B.1 Valores de aptitud utilizados

Aptitudes	Aptitudes Finales
0.96668	0.98921
0.96968	0.95981
0.98841	0.96138
0.98844	0.99985
0.9766	0.95723
0.96999	0.96787
0.97194	0.99985
0.98174	0.99985
0.98921	0.95723
0.98921	0.95981

A.5 Cálculo del coeficiente de Auto-correlación

Para calcular esta métrica, es necesario obtener el valor de aptitud promedio utilizando la expresión 1. El coeficiente de auto-correlación se calcula con la expresión 2.

(1)

$$\overline{f} = \frac{1}{n}\sum_{i=1}^{n} f, n > 0$$

$$\rho_k = \frac{\sum_{i=1}^{N-k}\left(f_i - \overline{f}\right)\left(f_{i+k} - \overline{f}\right)}{\sum_{i=1}^{N}\left(f_i - \overline{f}\right)^2}, n > k \qquad (2)$$

Calculando:

$\overline{f}$=0.97919

$$\rho_k = \frac{(0.9668\text{-}0.97919)(0.96968\text{-}0.97919)+....+(0.98921\text{-}0.97919)(0.98921\text{-}0.97919)}{(0.96668-0.97919)^2+...+(0.98921-0.97919)}$$

$$\rho_k = \frac{0.000290607}{0.00076872}$$

$$\rho_k = 0.37803805$$

Interpretación: El valor obtenido para el coeficiente de auto-correlación es cercano a 0 y esto indica que los valores de aptitud presentan poca correlación entre sí, a una distancia 1. Se puede decir que la superficie de solución es poco rugosa.

A.6 Cálculo de la longitud de auto-correlación (Autocorrelation lenght AC)

La longitud de auto-correlación se calcula utilizando la expresión 3.

$$\|\rho_k\| = \frac{1}{1-\rho_k} \qquad (3)$$

Aplicando esta expresión se obtienen:

$$|\rho_k| = \frac{1}{1 - 0.37802805}$$
$$|\rho_k| = 1.607815398$$

Interpretación: el valor obtenido para esta métrica es alto, y esto es indicador de una superficie poco rugosa.

A.7 Cálculo del Coeficiente de engaño

Para calcular esta métrica se utilizan los valores de aptitud finales obtenidos en varias ejecuciones del algoritmo.

$$coef\_eng = 1 - \frac{E - F_{mín}}{F_{máx} - F_{mín}} \tag{4}$$

Utilizando la expresión 4 y los valores de aptitud final mostrados en la Tabla 1 se obtiene el siguiente resultado.

$$F = \{0.98921, 0.95981, 0.96138, 0.99985, 0.95723,$$
$$0.96787, 0.99985, 0.99985, 0.95723, 0.95981\}$$
$$coef\_eng = 1 - \frac{0.975209 - 0.95723}{0.99985 - 0.95723}$$
$$coef\_eng = \frac{0.017979}{0.04262}$$
$$coef\_eng = 0.578155795$$

El valor obtenido se acerca a 1 por lo que se puede decir que el problema es engañoso.

A.8 Cálculo del contenido de información

Esta métrica se calcula generando primero la cadena *s*, a partir de los valores de aptitud por medio de la función presentada en la expresión 5.

$$\psi_{ft}(i,\varepsilon) = \begin{cases} \bar{1} & \text{si } f_i - f_{i-1} < -\varepsilon \\ 0 & \text{si } f_i - f_{i-1} \leq -\varepsilon \\ 1 & \text{si } f_i - f_{i-1} > -\varepsilon \end{cases} \tag{5}$$

Aplicando esta expresión se obtiene:

$$f = \{0.96668, 0.96968, 0.98841, 0.98844, 0.9766,$$
$$0.96999, 0.97194, 0.98174, 0.98921, 0.98921\}$$
$$s = \left\{1, 1, 1, \bar{1}, \bar{1}, 1, 1, 1, 0\right\}$$

La trayectoria asociada a esta cadena es la mostrada en la Figura B.1. Una vez obtenida la cadena *s*, se calculan las probabilidades de las posibles secuencias de elementos *pq* en *s* como se muestra en la Tabla B.2.

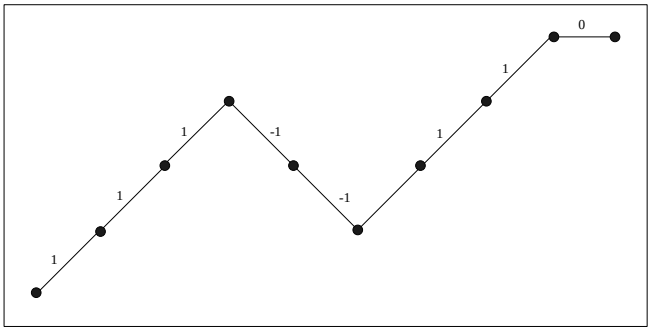


Figura B.1 Trayectoria de la cadena *s*

Tabla B.2 Probabilidad de las secuencias *pq* en *s*

[pq], p, q ∈ s	Ocurrencia([pq])	P([pq])
11	4	0.5
$1\bar{1}$	1	0.125
$\bar{1}\bar{1}$	1	0.125
$\bar{1}1$	1	0.125
10	1	0.125

Para calcular el contenido de la información que proporciona la cadena *s* se utiliza la expresión 6. En la Tabla B.3 se presentan las probabilidad de ocurrencia de las secuencias *pq* donde $p \neq q$ y también los cálculos de logaritmos requeridos.

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \qquad (6)$$

Tabla B.3 Probabilidad de ocurrencia de elementos pq, donde p≠q

[pq], p, q ∈ s, , p ≠ q	Ocurrencia([pq])	P([pq])	$P_{[pq]} \log_6 P_{[pq]}$
$1\bar{1}$	1	0.125	-0.145069803
$\bar{1}1$	1	0.125	-0.145069803
10	1	0.125	-0.145069803

Aplicando la expresión 6 se obtiene:

$$H(\varepsilon) = P_{[pq]} \log_6 P_{[pq]}$$
$$H(\varepsilon) = -\left(\text{-0.145069803} + \text{-0.145069803} + \text{-0.145069803}\right)$$
$$H(\varepsilon) = 0.435209408$$

Interpretación: el valor obtenido para esta métrica indica que la superficie de aptitudes no es muy rugosa

A.9 Cálculo de la información de la densidad de valles

Esta métrica se calcula con la expresión 7, los valores de probabilidad de ocurrencia de los bloques *pq* donde *p=q* y los cálculos de logaritmos se muestran en la Tabla B.4.

$$h(\varepsilon) = -\sum_{p \in \{\bar{1},0,1\}} P_{[pp]} \log_3 P_{[pp]} \qquad (7)$$

Tabla B.4 Probabilidades de ocurrencia de bloques *p=q*

[*pq*], *p*, *q* ∈ *s*, , *p* =*q*	P([*pq*])	$P_{[pq]} \log_6 P_{[pq]}$
11	0.5	-0.193426404
$\bar{1}\bar{1}$	0.125	-0.145069803

Entonces:

$$H(\varepsilon) = -\left(\text{-0.193426404} + \text{-0.145069803}\right)$$
$$H(\varepsilon) = 0.338496206$$

Interpretación: el valor encontrado para la métrica indica que no existen muchos valles de atracción en la trayectoria del algoritmo. En la Figura B.1 puede observarse que las secuencias no son largas y el algoritmo sale de ellas con relativa facilidad.

A.10 Cálculo del contenido parcial de la información y número esperado de óptimos

Para calcular esta métrica, es necesario calcular los cambios en la trayectoria con la expresión 8.

$$\phi_S(i,j,k) = \begin{cases} k & \text{si } i > n \\ \phi_S(i+1,i,k+1) & \text{si } j = 0 \text{ y } S_i \neq 0 \\ \phi_S(i+1,i,k+1) & \text{si } j > 0, S_i \neq 0 \text{ y } S_i \neq S_j \\ \phi_S(i+1,j,k) & \text{otro caso} \end{cases} \quad (8)$$

La Figura B.2 muestra los cambios que existen en la trayectoria de la cadena *S.* La Tabla B.5 muestra paso la aplicación de la expresión, tomando como punto de inicio $\mu = \phi_S(1,0,0)$. Donde el valor de k retorna la cantidad de cambios en la trayectoria.

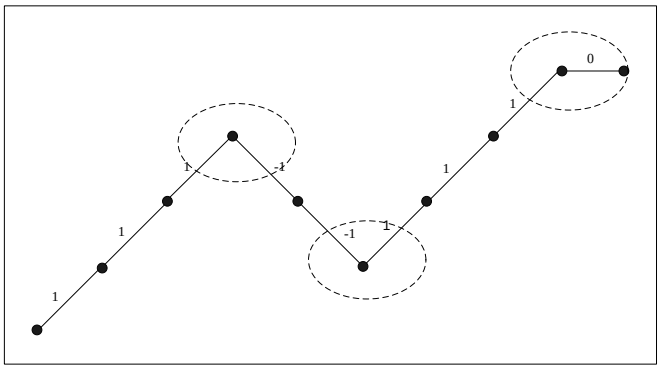


Figura B.2. Cambios en la trayectoria del algoritmo

Aplicando la expresión 9 se obtiene el valor del contenido parcial de la información.

$$M(\varepsilon) = \frac{\mu}{n} \quad (9)$$

El valor obtenido para *k* es 3, y *n* es la longitud de la cadena original que es 9, entonces, aplicando la expresión se tiene:

Tabla B.5 Cálculo de la cantidad de cambios en la trayectoria

i	*j*	*k*	Acción
1	0	0	$\phi_S(i+1,i,k+1)=\phi_S(2,1,1)$
2	1	1	$\phi_S(i+1,\ j,k)=\phi_S(3,1,1)$
3	1	1	$\phi_S(i+1,\ j,k)=\phi_S(4,1,1)$
4	1	1	$\phi_S(i+1,i,k+1)=\phi_S(5,4,2)$
5	4	2	$\phi_S(i+1,\ j,k)=\phi_S(6,4,2)$
6	4	2	$\phi_S(i+1,i,k+1)=\phi_S(7,6,3)$
7	6	3	$\phi_S(i+1,\ j,k)=\phi_S(8,6,3)$
8	6	3	$\phi_S(i+1,\ j,k)=\phi_S(9,6,3)$
9	6	3	$\phi_S(i+1,\ j,k)=\phi_S(10,6,3)$
10	6	3	$\mu=\phi_S(1,0,0)=3$

$$M(\varepsilon)=\frac{3}{9}$$
$$M(\varepsilon)=0.3333$$

Interpretación: el valor obtenido indica que no hay muchos cambios en la trayectoria, por lo que se puede considerar que la superficie no es rugosa.

El número esperado de óptimos en la trayectoria se calcula con la expresión 10

$$opt\_e=\frac{nM(\varepsilon)}{2} \qquad (10)$$

Entonces:

$$opt\_e=\frac{9*0.3333}{2}=1.4995\approx 2$$

Los óptimos locales son los puntos más altos en la trayectoria, como se puede observar en la Figura B.2, el número de óptimos locales es 2. Por lo que la estimación del número esperado de óptimos es correcta.

Anexo C

REDUCCIÓN DE DATOS USANDO ANALISIS DE FACTORES

El análisis por factores es una técnica que permite determinar si *p* variables respuesta exhiben patrones de relación entre si, tales que las variables se puedan dividir en m subconjuntos, en los que cada uno conste de un grupo de variables fuertemente correlacionadas.

Los subgrupos formados a partir del análisis de factores pueden considerarse como una reducción de la dimensionalidad de los datos originales y son un nuevo conjunto de variables no correlacionadas llamadas características subyacentes. Cuando se hace este análisis, se espera que las características subyacentes proporcionen una mejor comprensión de los datos originales y que éstas puedan ser utilizadas para análisis futuros.

C.1 Objetivos del análisis de factores

Los principales objetivos que se persiguen cuando se utiliza este tipo de análisis son:

a. Determinar si existe un conjunto mas pequeño de variables no correlacionadas que expliquen las relaciones existentes entre las variables originales
b. Determinar el número de factores subyacentes
c. Interpretar las nuevas variables

d. Convertir las observaciones originales a factores subyacentes

e. Usar las nuevas variables para análisis posteriores

C.2 Modelo para el análisis de factores

Suponga que se observa un vector *X* de respuestas, *p*-variado, de una población que tiene una media μ y matriz de varianzas-covarianzas Σ. En el modelo general del Análisis de factores se supone la existencia de *m* factores subyacentes (donde es deseable que $m < p$), denotados por $f_1, f_2, \ldots, f_m$ y un coeficiente único η_j en la forma:

$$x_j = \mu_j + \lambda_{j_1} f_1 + \lambda_{j_2} f_2 + \ldots \lambda_{jm} f_m + \eta_j \text{ Para } j=1, 2, \ldots, p.$$

Y se cumplen los siguientes supuestos:

1. Los f_k son independientes e idénticamente distribuidos, con media 0 y varianza 1, para $k = 1, 2, \ldots, m$.
2. Los η_j están independientemente distribuidos, con media 0 y varianza ψ_j, para $j = 1, 2 \ldots, p$.
3. f_k y η_j tiene distribuciones independientes para todas las combinaciones de *j* y *k*, donde $k = 1, 2, \ldots, m$; y $j = 1, 2 \ldots, p$.

Se pueden establecer las hipótesis de los f_k y los η_j tienen medias cero, sin perdida de generalidad. Esto se debe a que, si las medias no son ceros, su contribución podría describirse por las μ_j y los f_k y η_j tendrían medias cero. Asimismo, se puede establecer la hipótesis de los f_k tienen varianza 1, sin perdida de generalidad debido a que, si las varianzas no fueran todas iguales a 1, se podrían cambiar los λ para crear un nuevo conjunto de *F* que tendría varianza 1. Así la única restricción real en las hipótesis para el modelo del FA es que todos los *f* y η sean independientes.

Para simplificar el análisis siempre se supone que μ_j y que la varianza $\text{var}(x_j) = 1$, para todo *j*. Este siempre puede ser el caso, si simplemente se estandarizan las variables medidas antes de iniciar el análisis por factores y esta es la opción predeterminada en casi todos los paquetes para cálculos estadísticos. De donde, el modelo FA queda:

$$x_j = \lambda_{j_1 f_1} + \lambda_{j_2 f_2} + ... + \lambda_{j_m f_m} + \eta_j \quad \text{Para } j=1,2,\ldots p$$

En forma matricial el modelo es $X = \Delta f + \eta$ donde:

$$X = [x_1, x_2, ...x_p] \qquad \Delta = \begin{pmatrix} \lambda_{11} & \lambda_{12} & ...\lambda_{13} \\ \lambda_{21} & \lambda_{22} & ...\lambda_{2m} \\ \lambda_{p1} & \lambda_{p2} & ...\lambda_{pm} \end{pmatrix}$$
$$F = [f_i, f_2, ... f_m]$$
$$\eta = [\eta_1, \eta_2 ... \eta_p]$$

En la forma matricial, la hipótesis del modelo del análisis por factores queda de la siguiente manera:

1. $f \sim (0, I)$,
2. f □ (0,1), en donde ψ = diag ($\psi_{1,\cdots} \psi_{p,}$) y
3. f y η son independientes

C.3 Definiciones de la terminología del análisis por factores

Las nuevas variables $f_1, f_2, ... f_n$ se llaman factores comunes y los $\eta_1, \eta_2, ..., \eta_p$ se llaman factores específicos. La cantidad ψ_j describe la variación residual específica de la *j*_esima variable.

Los multiplicadores λ se llaman cargas de los factores. Cada λ_{jk} mide la contribución del *k*-esimo factor común a la *j*-esima variable respuesta y es la carga de la *j*-esima variable respuesta sobre el *k*-esimo factor.

C.4 Métodos para generar el análisis de factores

Existen varios métodos disponibles para resolver las ecuaciones del modelo de factores. La mayoría de estos métodos requieren que el usuario realice algunos juicios subjetivos, como inferir las comunidades para cada una de las variables respuesta.

1. Establecimiento de los factores principales, con o sin iteración.
2. Establecimiento de factores canónicos de Rao.
3. Establecimiento de los factores alfa.,
4. Establecimiento de los factores imágenes.
5. Máxima verosimilitud
6. Análisis por factores mediante mínimos cuadrados no ponderados.
7. Establecimiento de factores de Harris.

No se sabe cual de ellos es mejor. El método más popular es la extracción de factores a partir de los componentes principales. Dos ventajas que tiene el método de máxima verosimilitud sobre los otros son: 1) es invariante respecto a las unidades en las que se midan las variables y 2) existen maneras objetivas para ayudar al investigador a estimar un número apropiado de factores subyacentes. Si el investigador cree que los datos que se están analizando son normales multivariados, entonces se debe considerar seriamente el procedimiento de máxima verosimilitud.

C.5 Elección de la cantidad apropiada de factores

Para elegir la cantidad de factores se utilizan varios criterios, algunos métodos de extracción de factores requieren que se conozca de antemano el número de factores, una recomendación en este caso es generar un análisis preliminar de las Componentes Principales, con este análisis se tendrá una pista acerca de cuantos factores extraer. La conjetura inicial no siempre concordará con la determinación final. Sin embargo se debe partir de alguna cantidad de factores, para posteriormente seleccionar la cantidad de

factores que se tomaran en cuenta. Algunos criterios subjetivos para determinar el número de factores son los siguientes:

- No se incluyen factores triviales. Estos son aquellos que contengan solo una variable cargada principalmente sobre el mismo. El razonamiento se basa en que si un factor esta formado por una sola variable, no tiene caso utilizarlo si es posible utilizar la variable original.
- Puede utilizarse un criterio basado en considerar las diferencias entre las matrices de correlaciones de las variables originales y las obtenidas usando los factores. Si la diferencia es pequeña se puede intentar reducir el número de factores, en caso contrario, se considera incrementarlos.
- Una ayuda para identificar cuántos factores seleccionar, es el análisis de la gráfica scree y observar en que momento estabiliza el número de factores.

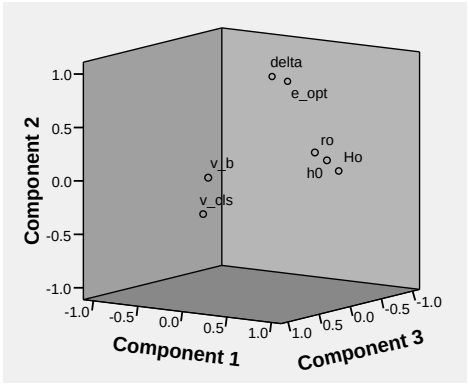
Anexo D

ANÁLISIS DE VERSIONES DEL ALGORITMO TABU

En este anexo se presenta el análisis de las versiones 2, 3, 4, 5, 6,7 y 8 del algoritmo Búsqueda Tabú. El proceso consta de dos etapas: análisis de factores para obtener los indicadores del proceso algorítmico y creación de modelos causales.

D.1 Análisis de versión 2

	Component		
	1	2	3
Ho	.924	.126	-.081
H0	.915	.251	.095
Ro	.859	.335	.209
delta	.217	.947	-.022
E_opt	.321	.899	-.121
V_b	.101	.114	.840
V_cls	.006	-.245	.784

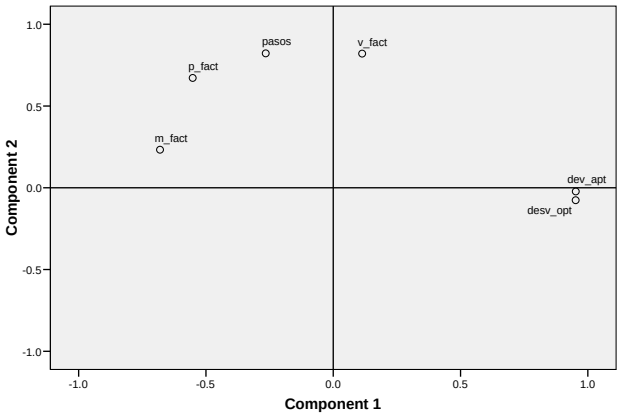


Componente	Nombre
1	Indicador de *rugosidad*
2	Indicador de *modalidad*
3	Indicador de *variabilidad*

Figura D.1 Indicadores de comportamiento de la versión 2

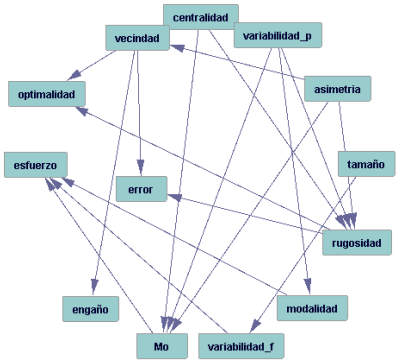
Las Figuras D.1 y D.2 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 2 para obtener los indicadores de comportamiento y desempeño. La Figura D.3 muestra el grafo causal generado para la versión 2 y las relaciones encontradas.

	Component	
	1	2
dev_apt	.953	-.022
desv_opt	.952	-.076
m_fact	-.680	.233
pasos	-.265	.822
v_fact	.113	.820
p_fact	-.552	.671



Componente	Nombre
1	*Error*
2	*esfuerzo*

Figura D.2 Indicadores del desempeño



Comportamiento←Complejidad
vecindad ← asimetría
variabilidad_f ← tamaño
Mo ← centralidad,variabilidad_p, asimetria
rugosidad ← centralidad, asimetría, varibilidad_p
modalidad ← variabilidad_p

Desempeño←Comportamiento
esfuerzo ← modalidad, Mo, variabilidad_f
error ← vecindad, rugosidad
engaño ← vecindad
optimalidad ← rugosidad, vecindad

Figura D.3. Grafo causal para la versión 2

En la Figura D.4 se presenta la estimación del grafo causal, y en las tablas D.1 y D.2 las ecuaciones estructurales asociadas al modelo.

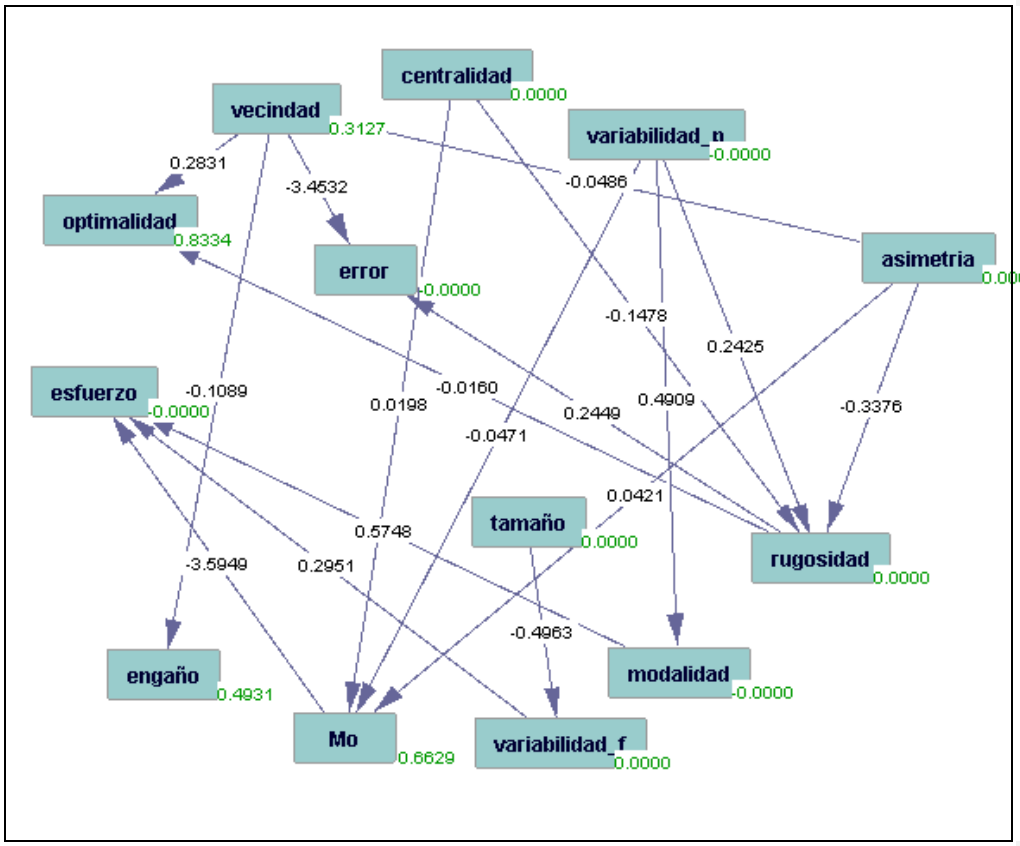


Figura D.4. Estimación del modelo causal para la versión 2

Tabla D.1 Ecuaciones estructurales para las relaciones Comportamiento← Complejidad de versión 2.

Comportamiento←Complejidad
vecindad = 0.3127 – 0.0486 *asimetria*
Mo = 0.6629 + 0.0198 *centralidad* – 0.0471 *variabilidad_p* +0.0421 *asimetria*
variabilidad_f = – 0.4963 *tamaño*
modalidad = 0.4909 *variabilidad_p*
rugosidad=-0.1478centralidad-0.3376asimetría+0.2425variabilidad_p

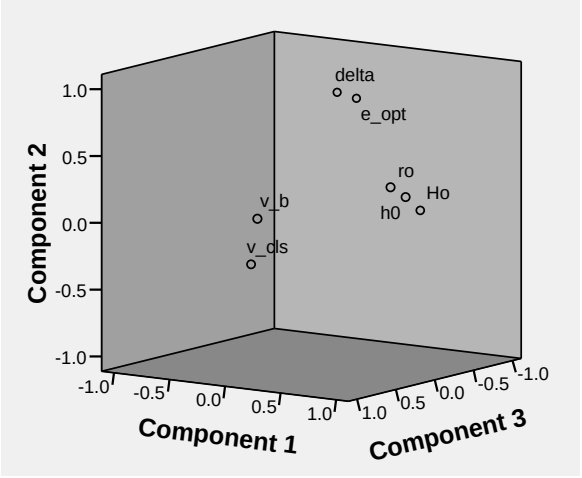
Tabla D.2 Ecuaciones estructurales para las relaciones Desempeño←Comportamiento de versión 2.

Desempeño←Comportamiento
optimalidad = 0.8334 + 0.2831 *vecindad* – 0.0160 *rugosidad*
error = – 3.4532 *vecindad* + 0.2449 *rugosidad*
esfuerzo = 0.5748 *modalidad* + 0.2951 *variabilidad_f* – 3.5949 *Mo*
engaño = 0.4931 – 0.1089 *vecindad*

D.2 Análisis de la versión 3

Las Figuras D.5 y D.6 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 3 para obtener los indicadores de comportamiento y desempeño.

	Component		
	1	2	3
Ho	.924	.126	-.081
h0	.915	.251	.095
ro	.859	.335	.209
delta	.217	.947	-.022
e_opt	.321	.899	-.121
v_b	.101	.114	.840
v_cls	.006	-.245	.784

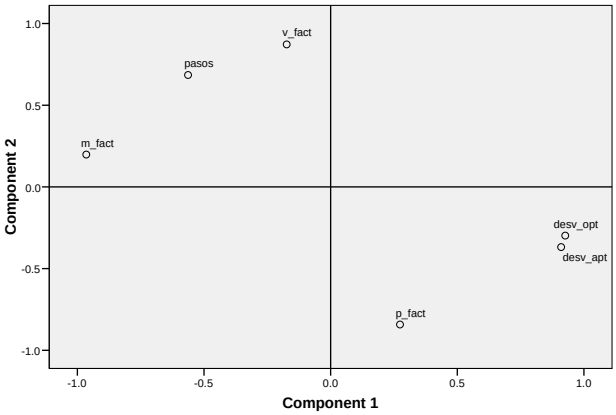


Componente	Nombre
1	*rugosidad*
2	*modalidad*
3	*variabilidad*

Figura D.5 Indicadores de comportamiento de la versión 3

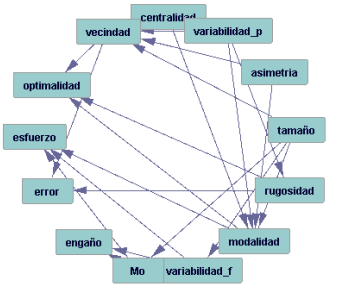
La Figura D.7 muestra el grafo causal estimado generado para la versión 3 y las relaciones encontradas.

	Component	
	1	2
m_fact	-.960	.214
desv_opt	.927	-.287
desv_apt	.921	-.353
var_fact	-.089	.898
pasos	-.463	.785
p_fact	.372	-.683



Componente	Nombre
1	*Error*
2	*esfuerzo*

Figura D.6 Indicadores del desempeño de la versión 3



Comportamiento←Complejidad
vecindad ←variabilidad_p, asimetría, tamaño *Mo ←tamaño,* *variabilidad_f ← tamaño* *modalidad ←tamaño, asimetría, variabilidad_p, centralidad* *rugosidad ←variabilidad_p*

Desempeño←Comportamiento
optimalidad ←rugosidad, modalidad *error ←vecindad, rugosidad* *engaño ← variabilidad_f, Mo* *esfuerzo ←variabilidad_f, Mo*

Figura D.7. Grafo causal para la versión 3

En la Figura D.8 se presenta la estimación del grafo causal, y en las tablas D.3 y D.4 las ecuaciones estructurales asociadas al modelo.

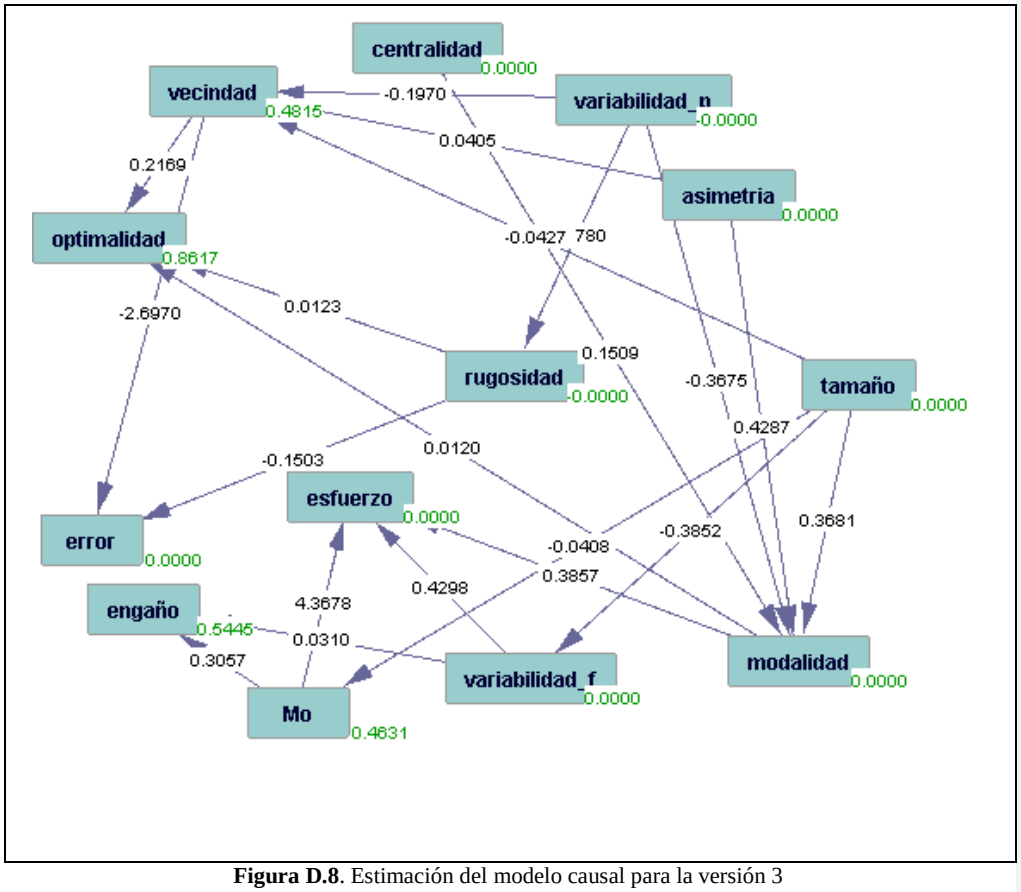


Figura D.8. Estimación del modelo causal para la versión 3

Tabla D.3 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 3.

Comportamiento←Complejidad
vecindad = 0.4815 – 0.1970 *variabilidad_p* + 0.0405 *asimetria* – 0.0427 *tamaño*
Mo = 0.4613 – 0.0408 *tamaño*
variabilidad_f = – 0.3852 *tamaño*
modalidad = 0.3681 *tamaño* + 0.4287 *asimetria* – 0.3675 *variablidad_p* – 0.1509 *centralidad*
rugosidad = – 0.1780 *varaiblidad_p*

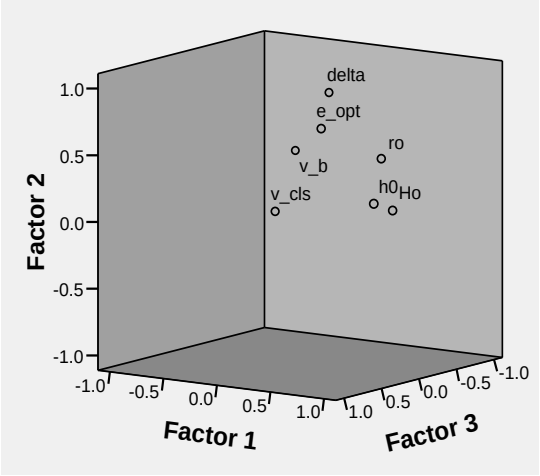
Tabla D.4 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 3.

Desempeño←Comportamiento
optimalidad = 0.8617 + 0.0123 *rugosidad* + 0.0120 *modalidad*
error = – 2.6970 *vecindad* – 0.1503 *rugosidad*
engaño = 0.5445 + 0.0310 *variabilida_f* + 0.3057 *Mo*
esfuerzo = 0.4298 *variabilidad_f* + 4.3678 *Mo*

D.3 Análisis de la versión 4

Las Figuras D.9 y D.10 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 4 para obtener los indicadores de comportamiento y desempeño.

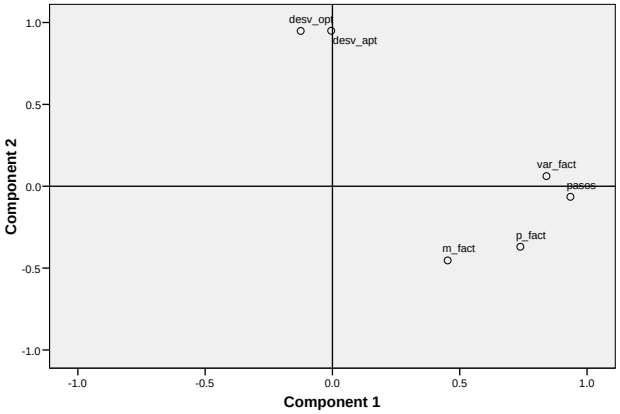
	Factor		
	1	2	3
Ho	.942	.149	.112
h0	.904	.224	.309
ro	.770	.507	.018
delta	.292	.956	.032
e_opt	.418	.739	.316
v_b	.063	.516	.155
v_cls	.107	.112	.486



Componente	Nombre
1	*Rugosidad*
2	*Modalidad*
3	*Variabilidad*

Figura D.9 Indicadores de comportamiento de la versión 4

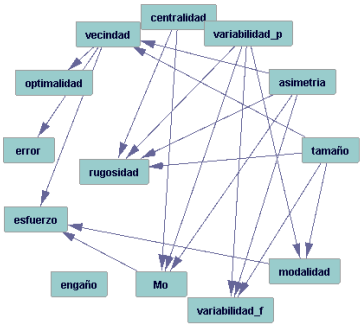
	Component	
	1	2
pasos	.935	-.065
var_fact	.841	.062
p_fact	.738	-.370
desv_apt	-.005	.950
desv_opt	-.125	.949
m_fact	.453	-.453



Componente	Nombre
1	*Esfuerzo*
2	*Error*

Figura D.10. Indicadores del desempeño de la versión 4

La Figura D.11 muestra el grafo causal estimado generado para la versión 4 y las relaciones encontradas.



Comportamiento←Complejidad
vecindad ← asimetría, tamaño
Mo ←centralidad, asimetría, variabilidad_p
variabilidad_f ← tamaño, variabilidad_p, asimetria
modalidad ←tamaño, variabilidad_p
rugosidad ←tamaño, variabilidad_p, asimetria

Desempeño←Comportamiento
optimalidad ←vecindad
error ←vecindad
esfuerzo ←vecindad, Mo, modalidad

Figura D.11. Grafo causal para la versión 4

En la Figura D.12 se presenta la estimación del grafo causal, y en las tablas D.5 y D.6 las ecuaciones estructurales asociadas al modelo.

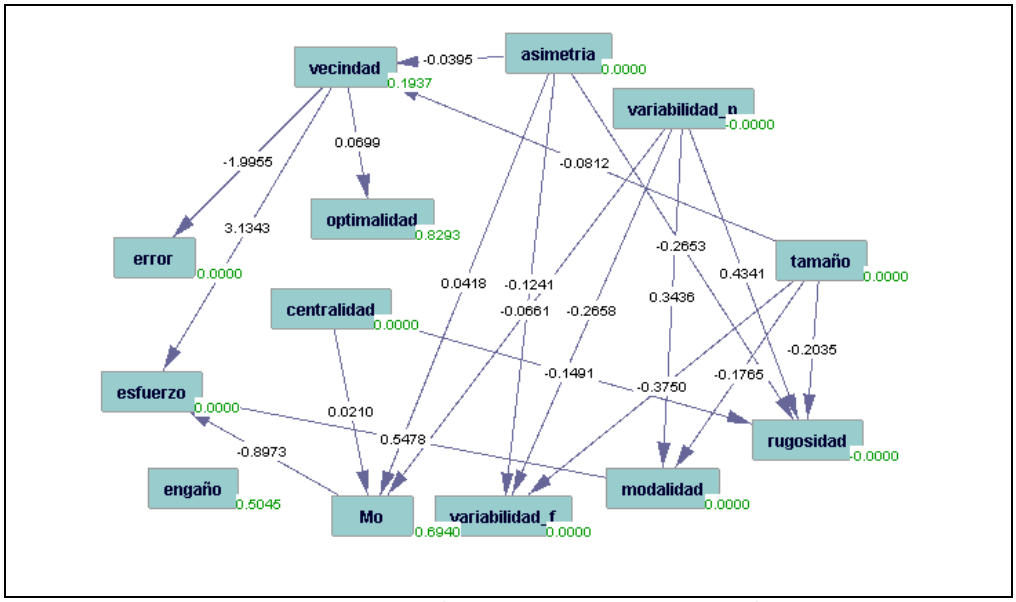


Figura D.12. Estimación del modelo causal para la versión 4

Tabla D.5 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 4.

Comportamiento←Complejidad
vecindad = 0.1937 – 0.0395 *asimetria* – 0.0812 *tamaño*
Mo = 0.6940 + 0.0210 *centralidad* – 0.0418 *asimetria* – 0.0661 *variablidad_p*
variabilidad_f = – 0.3750*tamaño* – 0.2658*variabilidad_p* – 0.1241*asimetria*
modalidad = – 0.1765 *tamaño* + 0.3436 *variabilidad_p*
rugosidad = 0.2035 *tamaño* + 0.4341*variabilidad_p* – 0.2653 *asimetria*

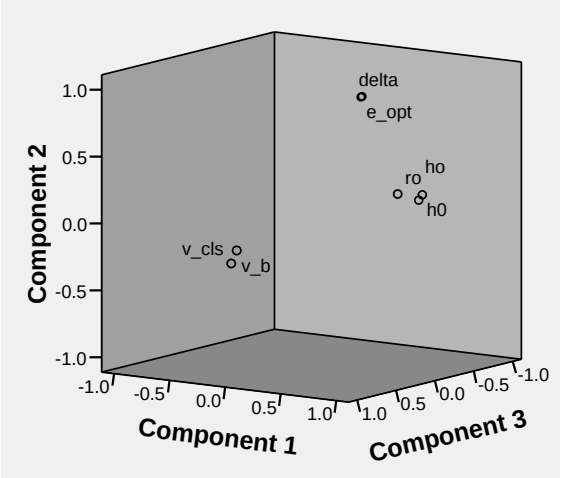
Tabla D.6 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 4.

Desempeño←Comportamiento
optimalidad = 0.8293 + 0.0699 *vecindad*
error = – 1.9955 *vecindad*
esfuerzo = 3.1343 *vecindad* – 0.8973 *Mo* + 0.5478 *modalidad*

D.4 Análisis de la versión 5

Las Figuras 13 y 14 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 5 para obtener los indicadores de comportamiento y desempeño.

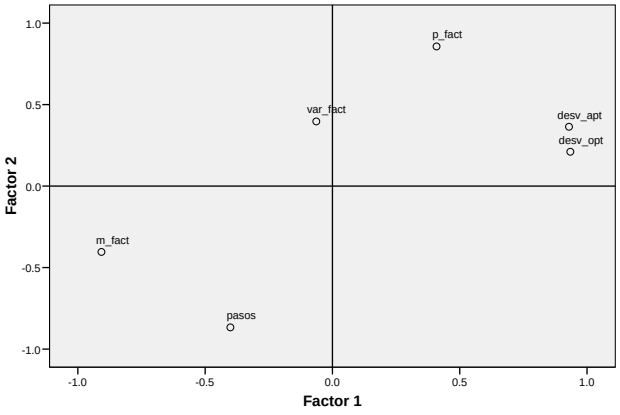
	Component		
	1	2	3
h0	.903	.238	-.135
ho	.857	.191	-.154
ro	.833	.268	.081
delta	.308	.900	-.196
e_opt	.294	.897	-.231
v_b	-.035	-.122	.910
v_cls	-.105	-.231	.882



Componente	Nombre
1	*Rugosidad*
2	*Modalidad*
3	*Variabilidad*

Figura D.13. Indicadores de comportamiento de la versión 5

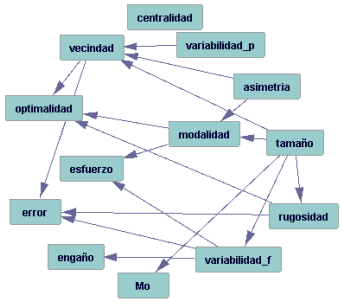
	Factor	
	1	2
desv_opt	.934	.211
desv_apt	.929	.364
m_fact	-.907	-.404
pasos	-.401	-.867
p_fact	.409	.857
var_fact	-.063	.397



Componente	Nombre
1	*error*
2	*esfuerzo*

Figura D.14. Indicadores del desempeño de la versión 5

La Figura D.15 muestra el grafo causal estimado generado para la versión 5 y las relaciones encontradas.



Comportamiento←Complejidad
vecindad ← variabilidad_p,asimetría, tamaño
variabilidad_f ←tamaño
modalidad ← tamaño, asimetria
rugosidad ←tamaño
Mo ←tamaño

Desempeño←Comportamiento o
optimalidad ←vecindad, rugosidad, modalidad
error ←vecindad, rugosidad, variabilidad_f
esfuerzo ←modalidad,variabilidad_f
engaño ←variabilidad_f

Figura D.15. Grafo causal para la versión 5

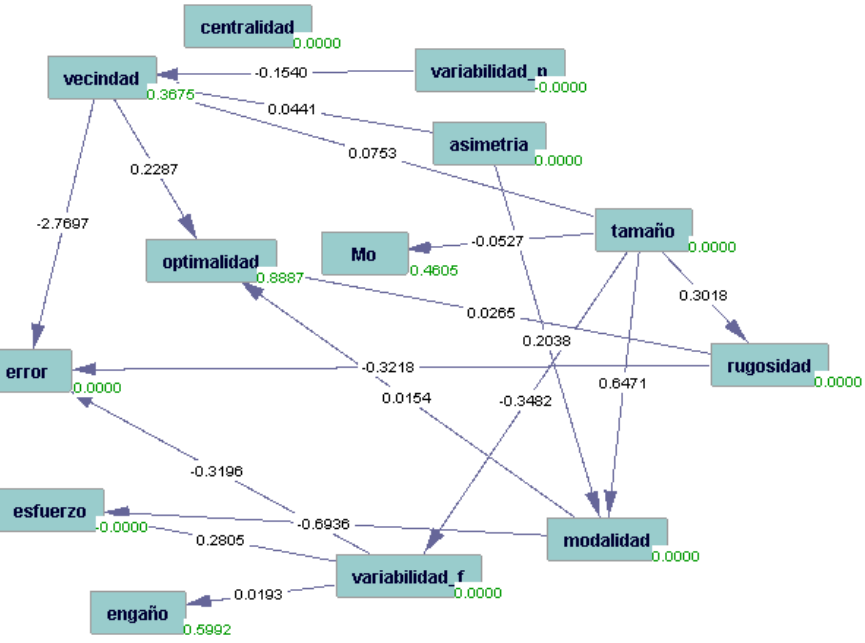


Figura D.16. Estimación del modelo causal para la versión 5

En la Figura 16 se presenta la estimación del grafo causal, y en las tablas 3 y 4 las ecuaciones estructurales asociadas al modelo.

Tabla D.7 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 5.

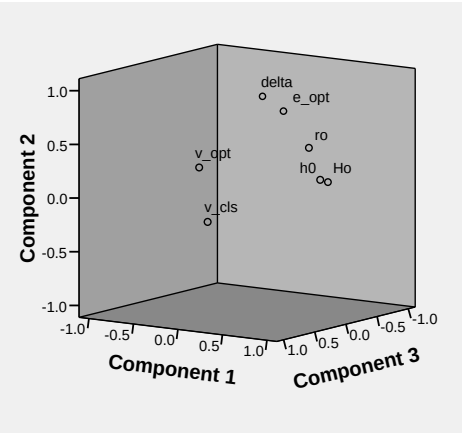
Comportamiento←Complejidad
vecindad = 0.3675 – 0.1540 *variabilidad_p* + 0.0441 *asimetria* + 0.0753 *tamaño*
variabilidad_f = – 0.3482 *tamaño*
modalidad = 0.6471 *tamaño* + 0.2083 *asimetria*
rugosidad = 0.3018 *tamaño*
Mo = 0.4607 – 0.0527 *tamaño*

Tabla D.8 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 5.

Desempeño←Comportamiento
optimalidad = 0.8887 + 0.2287 *vecindad* + 0.0265 *rugosidad* + 0.0154 *modalidad*
error = – 2.7697 *vecindad* – 0.3218 *rugosidad* – 0.3196 *variabilidad_f*
esfuerzo = – 0.6936 *modalidad* + 0.2805 *variabilidad_f*
engaño = 0.5992 + 0.0193 *variabilidad_f*

D.5 Análisis de la versión 6

	Component		
	1	2	3
Ho	.941	.202	.047
h0	.933	.239	.160
ro	.784	.517	.128
delta	.236	.936	.090
e_opt	.457	.817	.069
v_cls	.169	-.127	.874
v_opt	.018	.353	.795

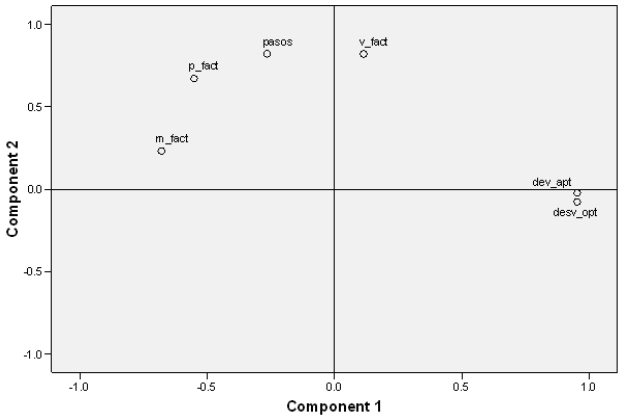


Componente	Nombre
1	*rugosidad*
2	*modalidad*
3	*variabilidad*

Figura D.17. Indicadores de comportamiento de la versión 6

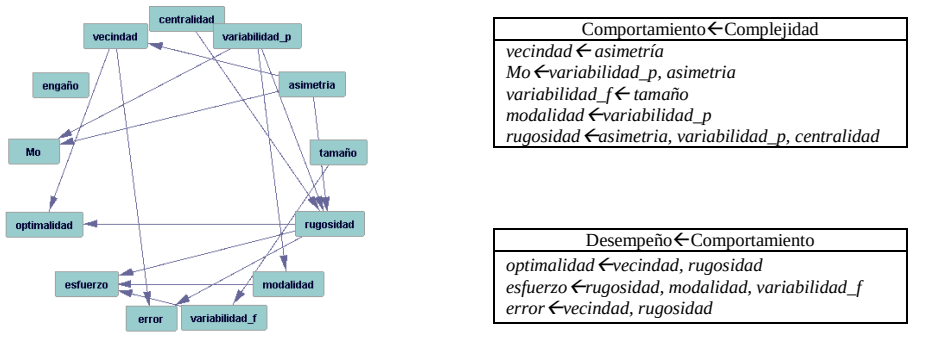
Las Figuras D.17 y D.18 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 6 para obtener los indicadores de comportamiento y desempeño.

	Component	
	1	2
dev_apt	.953	-.022
desv_opt	.952	-.076
m_fact	-.680	.233
pasos	-.265	.822
v_fact	.113	.820
p_fact	-.552	.671



Componente	Nombre
1	*Error*
2	*esfuerzo*

Figura D.18. Indicadores del desempeño de la versión 6



Comportamiento←Complejidad
vecindad ← asimetría
Mo ←variabilidad_p, asimetria
variabilidad_f ← tamaño
modalidad ←variabilidad_p
rugosidad ←asimetria, variabilidad_p, centralidad

Desempeño←Comportamiento
optimalidad ←vecindad, rugosidad
esfuerzo ←rugosidad, modalidad, variabilidad_f
error ←vecindad, rugosidad

Figura D.19. Grafo causal para la versión 6

La Figura D.19 muestra el grafo causal estimado generado para la versión 6 y las relaciones encontradas.

En la Figura D.20 se presenta la estimación del grafo causal, y en las tablas D.9 y D.10 las ecuaciones estructurales asociadas al modelo.

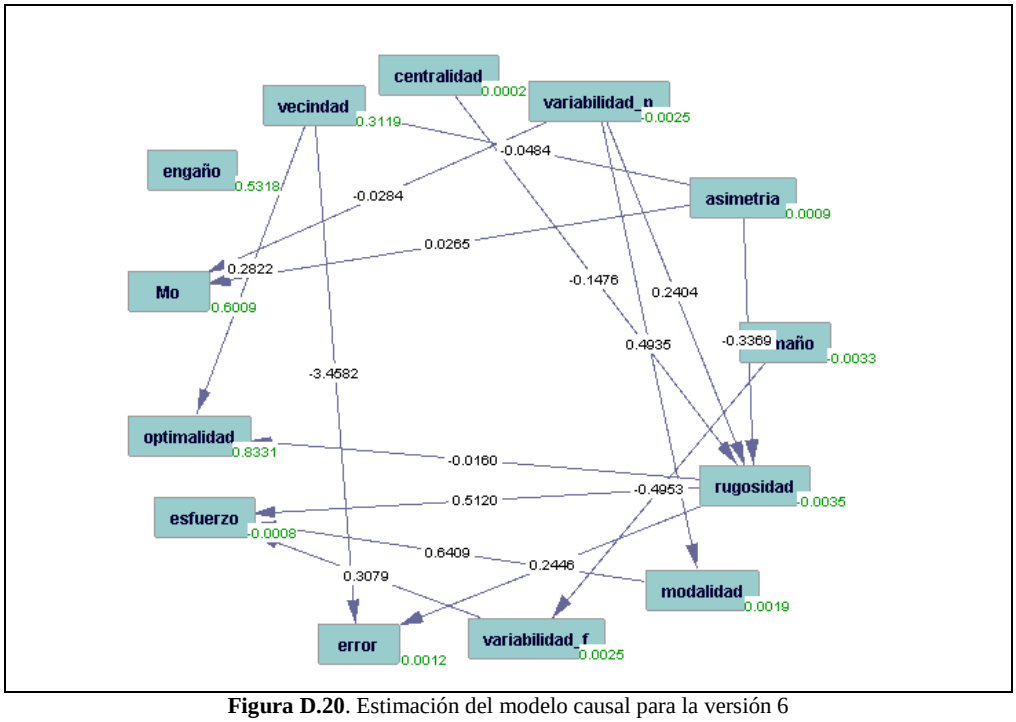


Figura D.20. Estimación del modelo causal para la versión 6

Tabla D.9 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 6.

Comportamiento←Complejidad
vecindad = 0.3119 – 0.0484 *asimetria*
Mo = 0.6009 – 0.0284 *variabilidad_p* + 0.0265 *asimetria*
variabilidad_f = 0.0024 – 0.4953 *tamaño*
modalidad = 0.0019 + 0.4936 *variabilidad_p*
rugosidad = – 0.0035 – 0.3369 *asimetria* + 0.2404 *variabilidad_p* – 0.1476 *centralidad*

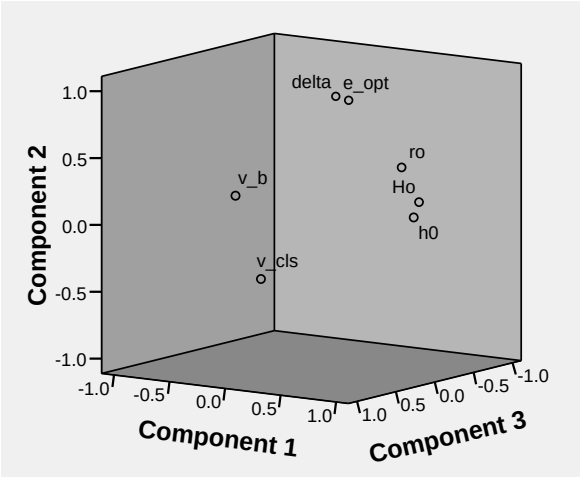
Tabla D.10 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 6.

Desempeño←Comportamiento
optimalidad = 0.8331 + 0.2822 *vecindad* – 0.0160 *rugosidad*
esfuerzo = 0.0008 + 0.5120 *rugosidad* + 0.6409 *modalidad* + 0.3079 *variabilidad_f*
error = 0.0012 – 3.4582 *vecindad* + 0.2446 *rugosidad*

D.6 Análisis de la versión 7

Las Figuras D.21 y D.22 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 7 para obtener los indicadores de comportamiento y desempeño.

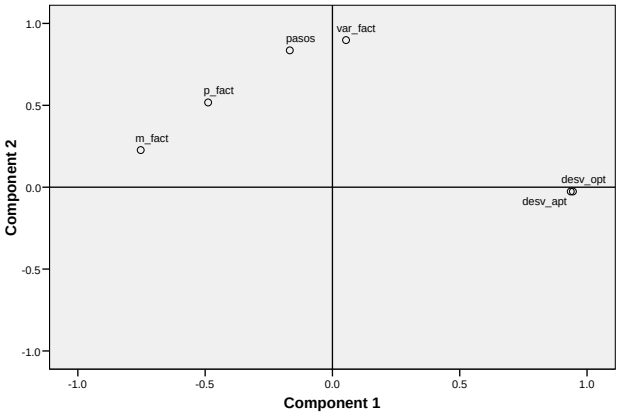
	Component		
	1	2	3
h0	.961	.113	.056
Ho	.922	.206	-.067
ro	.791	.457	-.032
delta	.223	.937	.005
e_opt	.320	.913	-.023
v_b	-.114	.276	.813
v_cls	.098	-.329	.788



Componente	Nombre
1	*Rugosidad*
2	*Modalidad*
3	*Variabilidad*

Figura D.21. Indicadores de comportamiento de la versión 7

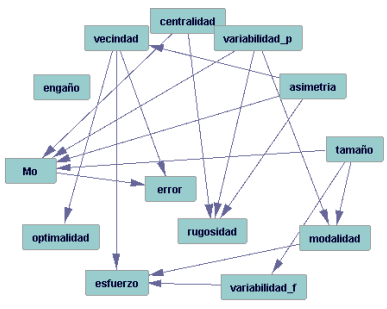
	Component	
	1	2
desv_opt	.944	-.026
desv_apt	.936	-.026
m_fact	-.753	.226
var_fact	.053	.898
pasos	-.168	.835
p_fact	-.488	.517



Componente	Nombre
1	*Error*
2	*esfuerzo*

Figura D.22 Indicadores del desempeño de la versión 7

La Figura D.23 muestra el grafo causal estimado generado para la versión 7 y las relaciones encontradas.



Comportamiento←Complejidad
vecindad ← asimetría
variabilidad_f ← tamaño
Mo ←tamaño, asimetría, variabilidad_p, centralidad
modalidad ←tamaño, variabilidad_p
rugosidad ←asimetria, variabilidad_p, centralidad

Desempeño←Comportamiento
optimalidad ←vecindad
esfuerzo ←vecindad, variabilidad_f, modalidad
error ←vecindad, Mo

Figura D.23 Grafo causal para la versión 7

En la Figura D.24 se presenta la estimación del grafo causal, y en las tablas 11 y 12 las ecuaciones estructurales asociadas al modelo.

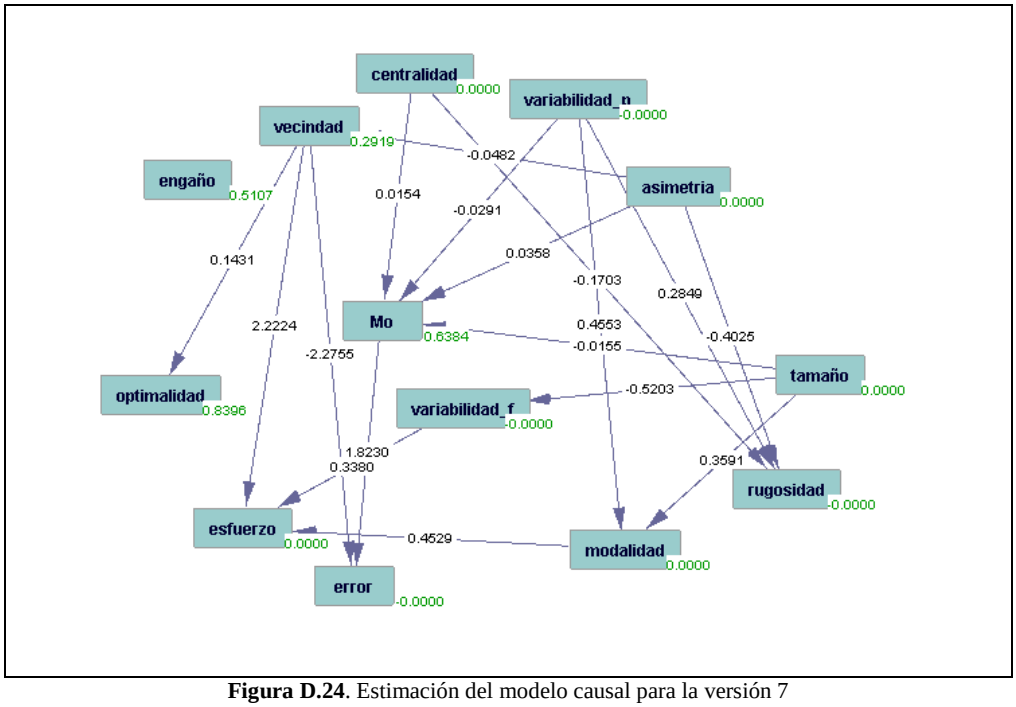


Figura D.24. Estimación del modelo causal para la versión 7

Tabla D.11 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 7.

Comportamiento←Complejidad
vecindad = 0.2919 – 0.0482 *asimetria*
variabilidad_f = – 0.5203 *tamaño*
Mo = 0.6384 – 0.0155 *tamaño* + 0.0358 *asimetria* – 0.0291 *variabilidad_p* + 0.0154 *centralidad*
modalidad = 0.3591 *tamaño* + 0.4553 *variabilidad_p*
rugosidad = – 0.4025 *asimetria* + 0.2849 *varaibilidad_p* – 0.1703 *centralidad*

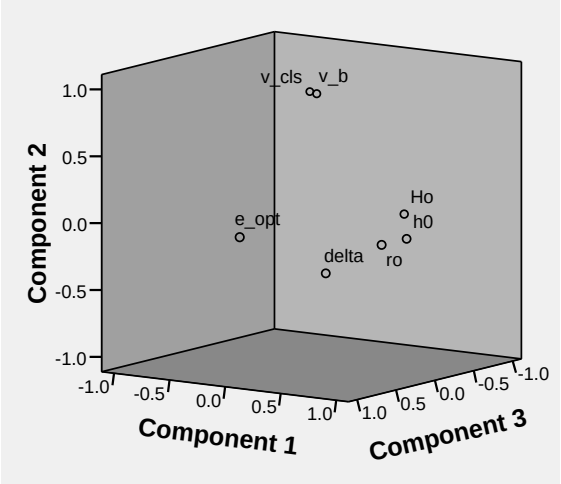
Tabla D.12 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 7.

Desempeño←Comportamiento
optimalidad = 0.8396 + 0.1431 *vecindad*
esfuerzo = 2.2224 *vecindad* + 0.3380 *variabilidad_f* + 0.4529 *modalidad*
error = - 2.2755 *vecindad* + 1.8230 *Mo*

D.7 Análisis de la versión 8

Las Figuras D.25 y D.26 presentan los resultados de aplicar el análisis de factores a las variables medidas de la versión 8 para obtener los indicadores de comportamiento y desempeño.

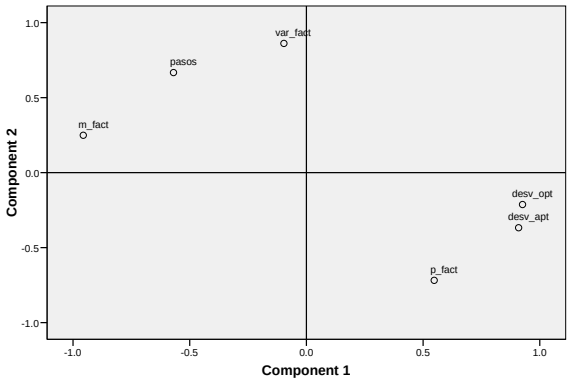
	Component		
	1	2	3
h0	.920	-.060	.091
ro	.804	-.093	.247
Ho	.749	.078	-.122
v_b	.024	.917	-.033
v_cls	-.084	.912	-.099
e_opt	.002	-.019	.925
delta	.482	-.302	.505



Componente	Nombre
1	*rugosidad*
2	*variabilidad*
3	*modalidad*

Figura D.25. Indicadores de comportamiento de la versión 8

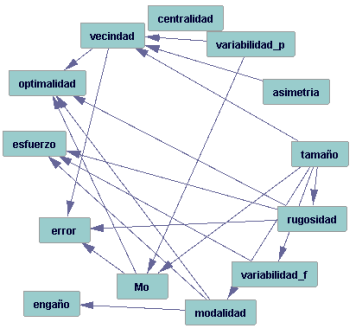
	Component	
	1	2
m_fact	-.956	.249
desv_opt	.926	-.213
desv_apt	.909	-.368
var_fact	-.096	.861
p_fact	.548	-.718
pasos	-.570	.667



Componente	Nombre
1	*error*
2	*esfuerzo*

Figura D.26 Indicadores del desempeño de la versión 8

La Figura D.27 muestra el grafo causal estimado generado para la versión 8 y las relaciones encontradas.



Comportamiento←Complejidad
vecindad ← variabilidad_p, asimetría, tamaño *modalidad ←tamaño* *variabilidad_f ← tamaño* *rugosidad ←tamaño*

Desempeño←Comportamiento
optimalidad ←vecindad, rugosidad, modalidad, Mo *esfuerzo ←rugosidad, variabilidad_f, modalidad* *error ←vecindad, rugosidad, Mo*

Figura D.27 Grafo causal para la versión 8

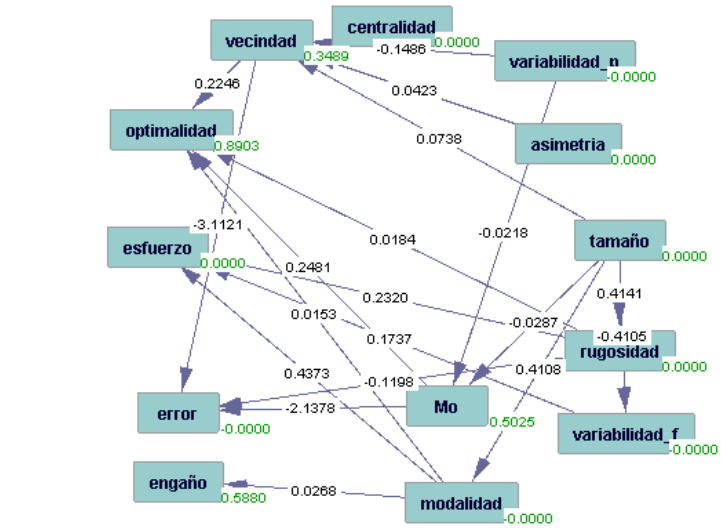


Figura D.28. Estimación del modelo causal para la versión 8

En la Figura D.28 se presenta la estimación del grafo causal, y en las Tablas D.13 y D.14 las ecuaciones estructurales asociadas al modelo.

Tabla D.13 Ecuaciones estructurales para las relaciones Complejidad→Comportamiento de versión 8.

Complejidad→Comportamiento
vecindad = 0.3489 – 0.1486 *variabilidad_p* + 0.0423 *asimetria* + 0.0738 *tamaño*
modalidad = 0.4108 *tamaño*
variabilidad_f = – 0.4105 *tamaño*
rugosidad = 0.4141 *tamaño*

Tabla D.14 Ecuaciones estructurales para las relaciones Comportamiento → Desempeño de versión 8.

Comportamiento→Desempeño
optimalidad = 0.8903 + 0.2246 *vecindad* + 0.0184 *rugosidad* + 0.0153 *modalidad* + 0.2481 *Mo*
esfuerzo = 0.2231 *rugosidad* + 0.1626 *variabilidad_ f* + 0.4092 *modalidad*
error = – 3.1121 *vecindad* – 0.1198 *rugosidad* – 2.1378 *Mo*

REFERENCIAS BIBLIOGRÁFICAS

[Agrawal98)] Agrawal, R., Gehrke J., Gunopulos D.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications.

[Álvarez06] Álvarez, H.: Modelo para Representar la Complejidad del Problema y el Desempeño de Algoritmos. Tesis Maestría. Instituto Tecnológico de Cd. Madero. 2004

[Barr95] Barr Richard S., Golden Bruce L., Kelly James P., Resendez M., Stewart William R. Designing and Reporting on Computational Experiments with Heurístic Methods. Journal of Heuristics, pp. 19-32

[Basse98] Basse, S.: Computer Algorithms, Introduction to Design and Analysis. Editorial Addison-Wesley Publishing Company .

[Beasley06] Beasley, J. E.: OR-Library. Brunel University. http ://people.brunel.ac.uk/~mastjjb/jeb/orlib/binpackinfo.html

[Borghetti96] Borghetti, B.J.: Inference Algorithm Performance and Selection under Constrained Resources. MS Thesis. FIT/GCS/ENG/96D-05.

[Bowes04] Bowes J., Neufeld E., Greer J., Cooke J. A Comparison of Association Rule Discovery and Bayesian Network Causal Inference Algorithms to Discover Relationships in Discrete Data.. Advances in Artificial Intelligence 13th Biennieal Conference of the Canadian Society for Computational Studies of Intelligence, AI 200, Montreal, Quebec, Canada, May 2000

[Brownle07] Brownle Jason,: A Note on Research Methodology and Benchmarking Optimization Algorithms ,Technical Report. Swinburne University of Technology (2007 Jan).

[Carnegie06]	Carnegie Mellon's University. Open Learning Initiative (OLI). http://www.cmu.edu/oli/
[Chickering95]	Chickering, D.: A transformational Characterization of Equivalent Bayesian Network Structures. 11 th Conference on Uncertanly AI. San Francisco, pp. 87-98
[Cofman02]	Coffman, J.E.G., Courboubetis, C., Garey, M.R., Johnson, D.S., Shor, P.W., Weber, R.R.: Perfect Packing Theorems and the Average Case Behavior of Optimal and Online *Bin Packing*. SIAM Review Vol. 44, pp. 95-108.
[Cohen95a]	Cohen, P.: Empirical Methods for Artificial Intelligence, The MIT Press Cambridge, Massachusetts, pp. 4-5, London, England, 1995.
[Cohen95b]	Cohen P., Gregory, D., Ballesteros, L., Amant. R.: Two algorithms for Inducing Structural Equation Models from Data. Computer Science Technical Report 94-80. Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statics, pp. 129-139 .
[Cruz04]	Cruz Reyes 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, Jun 2004).
[Ducatelle01]	Ducatelle, F., Levine, J.: Ant Colony Optimization for Bin Packing and Cutting Stock Problems. Proceedings of the UK Workshop on Artificial Intelligence. Edinburgh .
[EMVI07]	Enciclopedia Multimedia Interactiva y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas. 2007.
[Esposito 2000]	Esposito, F., aleaba, D., Ripa V., Semeraro G.: Discovering Causal Rules in Relational Databases. Applied Artificial Intelligence vol. 11 , pp. 71-84 (1997).
[Falkenauer96]	Falkenauer, E.: A Hybrid Grouping Genetic Algorithm for Bin Packing. Journal of Heuristics, Vol. 2, pp. 5-30.
[Fleszar02]	Fleszar, K., Hindi, K.: New Heuristics for One-Dimensional Bin Packing. Computers & Operations Research 29(7), pp. 821-839.

Fonloup97.	Fonloup C., Robilliard D., Preux P.: Landscape and the behavior of Heuristics, April 30.
[Glover86]	Glover, F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Comput. Oper. Res. Vol 13. (1986) 533-549
[Glymour&Cooper99]	Glymour C., Cooper, G. Computation, Causation & Discovery. AAAI Press/ The MIT Press. 1999
[Heckerman95]	Heckerman, D.: A Bayesian Approach to Learning Causal Networks. Reporte Técnico. MSR-TR-95-04. Microsoft Research. Advanced Technology Division. Microsoft Corporation.
[Hooker94]	J.N. Hooker, "Needed: An empirical science of algorithms", Operations Research, vol. 42, 1994.
[Hordijk05]	Hordijk W. An Introduction to Evolutionary Computation. K. Krithivasan and R. Rama (eds). Proceedings of the Research Level Group Discussion on Natural Computation, IIT Madras, Chennai, India. 2005
[Johnson02]	Johnson, D.:A Theoretician's Guide to the Experimental Analysis of Algorithms, en Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, M. H. Goldwasser, Editors, American Mathematical Society, Providence, pp. 215-250,
[Jones95a]	Jones F.: One operator, one landscape. Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA p. 95-02-025.
[Jones95b]	Jones, T.: Evolutionary Algorithms, Fitness Landscapes and Search. The University of New Mexico, Albuquerque, New Mexico. (1995)
[Kallel01]	Kallel L .; Naudts C.: Properties of Fitness Functions and Search Landscapes. Theoretical Aspects of Evolutionary Computing. 2001
[Lemeire04]	Lemeire, J., Dirkx, E.: Causal Models for Performance Analysis. 4th PA3CT Symposium (Septiembre 2004, Edegem, Belgica).

[Liu96]	Liu, H., Setiono, R.: A Probabilistic Approach to Feature Selection A filter solution. 13th International Conference on Machine Learning (ICML'96),1996, pp. 319-327. Bari, Italy.
[Liu98]	Liu, B, Hsu, W., y Ma, W.: Integrating classification and association rule mining. En Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 80-86.
[Loh06]	Loh Kok-Hua: Weight Annealing Heuristics for Solving Bin Packing and other Combinatorial Optimization Problems: Concepts, Algorithms, and Computational Results. Tesis Doctoral University of Maryland
[Lu02]	Lu, T., Druzdel, M.: Causal Models, Value of Intervention, and Search for Opportunities. Proceeding of the First European Workshop on Probabilistic Graphical Models (PGM-02) pp. 108-116, (Noviembre 2002, Cuenca España, 6-8)
[Madsen05]	Madsen, A., Jensen, F., Kjaerulff, U., Lang, M.: The Hugin Tool for Probabilistic Graphical Models. International Journal on Artificial Intelligence Tools 14(3), pp. 507-544 .
[Maes05]	Maes, S., Meganck, S., Manderick, B.: Identification of Causal Effects in Multi-agent Causal Models. Preceedings Artificial Intelligence and Applications .
[McGeoch00]	McGeoch, C.C.: Experimental Analysis of Algorithms. In: Pardalos, P.M., Romeijn, H.E.: Handbook of Global Optimization, Vol. 2, pp. 489-513 .
[Merz98]	Merz Peter Freisleben Bernd.: Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. Scientific literature Digital Library.
[Merz04]	Mertz P.: Advanced Fitness Landscape Analysis and the Performance of Memetic Algorithms, Vol. 12 pp: 303 – 325
[Mitchell97]	Mitchell M.T, Machine Learning. Ed. Mc Grow Hill, 1997
[Montgomery04]	Montgomery, Douglas. Diseño y Análisis de Experimentos. Limusa Willey. Segunda Edición .

[Norsys06]	Norsys Software Corp. Netica, Bayesian Network Development Software .
[Papadimitriou98]	Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Publications .
[Pearl00]	Pearl, J. Causality.: Models, Reasoning and Inference. Cambridge University Press .
[Pearl02]	Pearl, J.: Reasoning with cause and effect. IA magazine, Vol. 23, pp. 95-111 (Marzo 2002).
[Pearl03]	Pearl, J. Statics and Causal Inference: A review. Sociedad Española de Estadística e Investigación Operativa (TEST). Vol. 12, pp. 281-345 (Diciembre de 2003).
[Pérez04]	Pérez, J., Pazos, R.A., Frausto, J., Rodríguez, G., Cruz, L., Fraire, H.: Comparison and Selection of Exact and Heuristic Algorithms. Lectures Notes in Computer Science, Vol. 3045. Springer Verlag, Berlin Heidelberg New York, 415-424
[Quinlan93]	Quinlan, J. R. 1993. C4.5: Programs for machine learning. San Mateo, Calif.: Morgan Kaufmann.
[Reidys02]	Reidys C., Stadler Peter F**.:** Combinatorial Landscapes Vol 44 pp. 3 - 54
[Russel04]	Russel, S., Norving, P.: Inteligencia Artificial. Un enfoque moderno. Prentice Hall Hispanoamericana .
[Seattle06]	Seattle.: Analysis of the difficulty of learning goal-scoring behavior for robot soccer. Proceedings of the 8th annual conference on Genetic and evolutionary computation. pp. 1569 - 1576 .(Washington, USA) .
[Smith02]	Smith, T., Husbands, P., Layzell, P., O´Shea, M.: Fitness Landscapes and Evolvability. Centre for Computational Neuroscience and Robotics, School of Biological Sciences, University of Sussex, Brighton, UK (2002)
[Soares00]	Soares, C., Brazdil, P.: Zoomed Ranking, Selection of Classification Algorithms Based on Relevant Performance Information.Principles of Data Mining in Knowledge Discovery. Lecture Notes on Artificial Intelligence Vol. 1910.

	pp. 126-135, Springer Verlag, Berlin Heidelberg New York (2000)
[Spirtes00]	Spirtes, P.: An Anytime Algorithm for Causal Inference. citeseer.ist.psu.edu/385601.html
[Spirtes2001]	Spirtes, P., Glymour, C., Scheines, R. Causation, Prediction, and Search. MIT Press, 2nd edition (2001).
[Vassilev99]	Vassilev Vesselin K.,Julian F. Miller,: Digital Circuit Evolution: The Ruggedness and Neutrality of Two-Bit Multiplier Landscapes. IEE Half-day Colloquium on Evolutionary Hardware Systems.
[Verma91]	Verma, TS., Pearl, J.: Equivalence and Synthesis of Causal Models. Uncertanly in Artificial
[Yin03]	Yin, Xiaoxin y Han, Jiawei,: CPAR: Classification Based on Predictive Association Rules, SIAM International Conference on Data Mining.
[Weinberg90]	Weinberg, E. D. Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cibernetics, 63, 325-336. 1990
[Witten00]	Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers
[Wright32]	Wright, S. The roles of mutation, inbreeding, crossbreeding and selection in evolution. Proceedings of the sixth international genetics, vol. 1 pp. 356-366

CURRICULUM VITAE

Nombre completo:

Verónica Pérez Rosas

Lugar y fecha de nacimiento:

Álamo, Veracruz, México. 11 de Julio de 1981

Institución, duración, grado:

ITCM	1999-2004	Ingeniero en Sistemas Computacionales
CETIS 109	1996-1999	Técnico en Computación Fiscal Contable

Publicaciones:

Análisis Causal del Algoritmo Aceptación por Umbral, con Laura Cruz R., Vanesa Landero N., Joaquín Pérez O., Rodolfo A. Pazos. R. 14th International Congress on Computer Research CIICC'7. 2007.

Explaining Performance of the Threshold Accepting Algorithm for the Bin Packing Problem: A Causal Approach, con J. Pérez, L. Cruz, V. Landero, R. Pazos y G. Zarate. ACS. Advanced Computer Systems. Special volume of the Polish Journal of Environmental Studies. 2007

Búsqueda Supervisada para Recocido Simulado, con Santiago Gómez Carpizo, Guadalupe Castilla Valdez, Héctor Fraire Huacuja. 7mo. Simposium Iberoamericano de Computación e Informatica. 2006

Agente de Aprendizaje para Representar la Complejidad del Problema y Desempeño de Algoritmos, con Cruz R. L., Gómez S. C., Landero N. V., Alvarez H., V. 13th International Congress on Computer Research CIICC'6, pag 23-33. 2006.

An Ordered Preprocessing Scheme for Data Mining, con Laura Cruz R., Joaquín Pérez, Vanesa Landero, Elizabeth S. Del Angel, Victor M. Alvarez. PRICAI 2004: Trends in Arficial Intelligence. Lecture Notes in Computer Science. Springer Berlin/Heidelberg. 2004.