

11-100043-001

20. 2023



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Laboratorio de Tecnologías de Información,
CINVESTAV-Tamaulipas

**Optimización numérica mediante
algoritmos de estimación de
distribución**

Tesis que presenta:

Mario Rogelio Flores Méndez

Para obtener el grado de:

**Maestro en Ciencias
en Computación**

Director de la Tesis:
Dr. Ricardo Landa Becerra

Cd. Victoria, Tamaulipas, México.

Octubre, 2012

**CINVESTAV
IPN
ADQUISICION
LIBROS**

| | |
|----------|---------------|
| CLASIF.. | UT 00048 |
| ADQUIS.. | UT-T00048-SS |
| FECHA: | 11-Julio-2013 |
| PROCED.. | Dom 2013 |
| | \$ |

12:202015-201

© Derechos reservados por
Mario Rogelio Flores Méndez
2012

La tesis presentada por Mario Rogelio Flores Méndez fue aprobada por:

Dr. Eduardo Rodríguez Tello

Dr. Gregorio Toscano Pulido

Dr. Ricardo Landa Becerra, Director

Cd. Victoria, Tamaulipas, México., 30 de Octubre de 2012

Índice General

| | |
|---|-------------|
| Índice General | I |
| Índice de Figuras | v |
| Índice de Tablas | vii |
| Índice de Algoritmos | ix |
| Resumen | xi |
| Abstract | xiii |
| 1. Introducción | 1 |
| 1.1. Antecedentes y motivación para el proyecto | 1 |
| 1.2. Planteamiento del problema | 3 |
| 1.3. Objetivos generales y específicos del proyecto | 5 |
| 1.3.1. Generales | 5 |
| 1.3.2. Específicos | 5 |
| 1.4. Organización de la tesis | 6 |
| 2. Antecedentes | 7 |
| 2.1. Algoritmos evolutivos | 7 |
| 2.2. Marginales | 9 |
| 2.2.1. Distribución Beta | 9 |
| 2.2.1.1. Función de densidad de probabilidad | 9 |
| 2.2.1.2. Función de distribución acumulada | 9 |
| 2.2.1.3. Formas | 11 |
| 2.2.1.4. Estimación de parámetros | 12 |
| 2.2.1.5. Generación de variables aleatorias | 12 |
| 2.3. Cópulas | 12 |
| 2.3.1. Cópulas arquimedeanas | 13 |
| 2.3.1.1. Cópula de Frank | 15 |
| 2.3.1.2. Cópula de Gumbel | 15 |
| 2.3.1.3. Cópula de Clayton | 15 |
| 2.4. Modelos gráficos probabilísticos | 18 |
| 2.4.1. Redes bayesianas | 18 |
| 2.4.2. Redes Gaussianas | 20 |
| 2.4.3. Vines | 20 |
| 2.4.4. Métodos de inducción | 20 |

| | | |
|-----------|---|-----------|
| 2.4.4.1. | Métodos de exclusión de aristas. | 21 |
| 2.4.4.2. | Métodos score+search. | 21 |
| 2.4.5. | Estimación de la máxima verosimilitud. | 21 |
| 2.5. | Algoritmos de estimación de distribución. | 21 |
| 2.5.1. | Funcionamiento | 22 |
| 2.5.2. | EDAs en optimización numérica | 23 |
| 2.5.2.1. | Discretización | 23 |
| 2.5.2.2. | Representación directa | 24 |
| 2.6. | Resumen | 30 |
| 3. | Enfoque propuesto | 33 |
| 3.1. | Selección de individuos | 34 |
| 3.2. | Tau de Kendall | 34 |
| 3.3. | Estimación de parámetros | 35 |
| 3.3.1. | Estimación incremental de parámetros | 36 |
| 3.4. | Modelo gráfico probabilístico | 37 |
| 3.4.1. | Red bayesiana de cópulas | 38 |
| 3.4.2. | Funciones de densidad | 40 |
| 3.4.2.1. | Cópula de Frank | 41 |
| 3.4.2.2. | Cópula de Gumbel | 42 |
| 3.4.2.3. | Cópula de Clayton | 42 |
| 3.4.3. | Aprendizaje de la estructura del grafo | 43 |
| 3.4.4. | Estimación de parámetros finales de las cópulas | 47 |
| 3.5. | Muestreo | 47 |
| 3.6. | Resumen | 48 |
| 4. | Resultados y evaluación | 51 |
| 4.1. | Implementación | 51 |
| 4.2. | Experimentos | 53 |
| 4.3. | Experimentos variando el parámetro de relajación | 54 |
| 4.4. | Experimentos utilizando el calculo automático del parámetro de relajación | 61 |
| 4.4.1. | Experimentos con 10 variables | 62 |
| 4.4.2. | Experimentos con 50 variables | 69 |
| 4.5. | Resumen | 72 |
| 5. | Conclusiones y trabajo futuro | 73 |
| 5.1. | Conclusiones | 73 |
| 5.2. | Trabajo futuro | 75 |
| A. | Anexo | 77 |
| | Bibliografía | 81 |

Índice de Figuras

| | |
|---|----|
| 2.1. Función de densidad de probabilidad | 10 |
| 2.2. Función de distribución acumulada | 10 |
| 2.3. Cópula de Frank | 16 |
| 2.4. Cópula de Gumbel | 17 |
| 2.5. Cópula de Clayton | 19 |
| 3.1. Red bayesiana de cópulas | 40 |
| 4.1. Función 1 | 56 |
| 4.2. Función 2 | 57 |
| 4.3. Función 3 | 59 |
| 4.4. Función 19 | 60 |
| 4.5. Función 20 | 61 |
| 4.6. Funciones separables | 63 |
| 4.7. Funciones con un grupo no separable de tamaño m | 65 |
| 4.8. Funciones con $\frac{D}{2^m}$ grupos no separables de tamaño m | 67 |
| 4.9. Funciones con $\frac{D}{m}$ grupos no separables de tamaño m | 68 |
| 4.10. Funciones no separables | 70 |

Índice de Tablas

| | |
|--|----|
| 4.1. Función 1 | 55 |
| 4.2. Función 2 | 57 |
| 4.3. Función 3 | 58 |
| 4.4. Función 19 | 59 |
| 4.5. Función 20 | 60 |
| 4.6. Funciones separables | 62 |
| 4.7. Funciones con un grupo no separable de tamaño m | 64 |
| 4.8. Funciones con $\frac{D}{2^m}$ grupos no separables de tamaño m | 66 |
| 4.9. Funciones con $\frac{D}{m}$ grupos no separables de tamaño m | 66 |
| 4.10. Funciones no separables | 69 |
| 4.11. Funciones con un grupo no separable de tamaño m | 69 |
| 4.12. Funciones con $\frac{D}{2^m}$ grupos no separables de tamaño m | 71 |
| 4.13. Funciones con $\frac{D}{m}$ grupos no separables de tamaño m | 71 |
| 4.14. Funciones no separables | 71 |

Índice de Algoritmos

| | | |
|----|--|----|
| 1. | Búsqueda de grafo condicional | 45 |
| 2. | Encontrar arista por métrica con propuesta | 46 |

Optimización numérica mediante algoritmos de estimación de distribución

por

Mario Rogelio Flores Méndez

Maestro en Ciencias del Laboratorio de Tecnologías de Información, CINVESTAV-Tamaulipas
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2012
Dr. Ricardo Landa Becerra, Director

Los algoritmos de estimación de distribución son técnicas basadas en los algoritmos evolutivos, pero a diferencia de éstos, utilizan modelos probabilísticos para muestrear nuevos individuos en lugar de utilizar operadores de variación. El principal problema de este tipo de algoritmos es que pueden converger de manera prematura, sobre todo si las funciones objetivo son multi-modales. En este trabajo de tesis se presenta un algoritmo de estimación de distribución que utiliza una red bayesiana construida con cópulas arquimedeanas como modelo probabilístico. Se usan cópulas arquimedeanas de un parámetro para modelar las dependencias entre variables y la distribución beta es usada para describir los marginales univariados. El proceso de aprendizaje de la red bayesiana es ayudado por una técnica simple que se apoya de la propiedad asociativa de las cópulas arquimedeanas, el uso del tau de Kendall para medir la relación entre dos variables y la relación del tau de Kendall y las cópulas arquimedeanas de dos variables.

Numerical optimization through estimation of distribution algorithms

by

Mario Rogelio Flores Méndez

Master of Science from the Information Technology Laboratory, CINVESTAV-Tamaulipas
Research Center for Advanced Study from the National Polytechnic Institute, 2012
Dr. Ricardo Landa Becerra, Advisor

Estimation of distribution algorithms are techniques based on evolutionary algorithms with the singularity that, instead of using variation operators, they use probabilistic models to generate new individuals. The main problem with this kind of algorithms is that they may converge prematurely, specially when optimizing multi-modal functions. An estimation of distribution algorithm that employs a copula bayesian network as probabilistic graphic model, is presented. Multivariate archimedean copula functions with one parameter are used to model dependencies between variables and the beta distribution is used to describe the univariate marginals. The learning process of the bayesian network is assisted through a simple technique that relies on the associative property of archimedean copulas, the use of Kendall's tau for measuring relation between variables and the relation between Kendall's tau and bivariate archimedean copulas.

1

Introducción

1.1 Antecedentes y motivación para el proyecto

En el mundo real frecuentemente se presentan problemas de optimización numérica y combinatoria, que pueden ser resueltos por técnicas clásicas, como el método simplex, programación dinámica, búsqueda aleatoria, método del gradiente conjugado, método de ramificación y poda, etc. Pero existen algunos problemas que, debido a su naturaleza, no son diferenciables, no están completamente definidos por funciones analíticas o simplemente no pueden ser atacados por este tipo de técnicas debido a alguna dificultad.

Los algoritmos evolutivos son técnicas generales inspiradas en el principio de selección natural y basadas en población, que pueden lidiar con muchos problemas que no pueden ser resueltos por técnicas clásicas, sin importar si el problema presenta multi-modalidades, discontinuidades, restricciones, funciones ruidosas, etc. Un gran problema de estas técnicas es que dependen de varios parámetros como el tamaño de la población, los operadores de variación y las probabilidades de que estos operadores sean aplicados sobre los individuos. Si no se cuenta con conocimiento previo del problema específico y de los parámetros correctos para atacar el problema, la elección misma de los

parámetros se puede convertir en un gran problema y puede afectar en la convergencia del algoritmo y en la obtención de buenas soluciones. Otro problema de este tipo de técnicas es que son estocásticas, por lo tanto, su desempeño y los movimientos de la población en el espacio de búsqueda son muy difíciles de predecir.

Se ha demostrado que tomar en cuenta las relaciones entre las variables de decisión de un problema puede ayudar a la convergencia más rápida de los algoritmos evolutivos [Hansen and Ostermeier, 2001], lo que se ha atacado de diferentes formas en ciertas propuestas como la agrupación de variables aleatoriamente, por ejemplo [Omidvar et al., 2010]. Esta es una de las motivaciones originales de los algoritmos de estimación de distribución, pues intentan modelar explícitamente las relaciones entre las variables mediante modelos probabilísticos a partir de los cuales se muestrean nuevos individuos. El uso de estos modelos elimina la necesidad de operadores de variación y, por lo tanto, el problema de elegir los parámetros correctos para este tipo de operadores para cada problema en específico.

La manera en la que se agrupan las variables o se describen mediante un modelo gráfico es de gran importancia, pues de éste depende la forma en que se representarán las relaciones entre las variables. Esta representación tiene gran impacto en el tipo de problemas que se pueden atacar eficientemente por cada propuesta, pues la complejidad de las relaciones entre las variables en cada problema es variable y, por ejemplo, un modelo muy complicado puede degradar la convergencia de un algoritmo cuando se intenta resolver un problema en el cual la relación entre las variables es nula o muy débil, lo cual también puede ocurrir de manera inversa, donde un modelo que no toma en cuenta las relaciones o las agrupa en pares no ayudará al algoritmo en cuestión a converger eficientemente cuando se aplica a un problema donde las relaciones son más complejas. La forma tradicional de tratar este problema es el método de prueba y error, donde primero se utiliza un algoritmo sin dependencias, después uno con dependencias de dos variables y por último uno con dependencias multi-variables, hasta que se obtenga un resultado lo suficientemente bueno. Esto se trata de resolver algunos trabajos recientes, sin embargo aun no se tienen resultados lo suficientemente buenos, pues no existen propuestas que hagan éstas agrupaciones de manera incremental, lo que hace que esta etapa sea costosa pues se tiene que hacer la agrupación de variables en cada iteración, además que se dificulta cada vez más

según crezca la dimensionalidad del problema.

Los algoritmos de estimación de distribución se han apoyado en la distribución normal para representar la distribución de las regiones del espacio de búsqueda donde se encuentran óptimos, lo que hace que se aumente la velocidad de convergencia debido a que en cada generación la distribución de probabilidad se sesga cada vez más en torno a un valor medio y, por lo tanto, las soluciones generadas cada vez serán más cercanas entre sí. Esto es un problema en funciones multimodales, pues el algoritmo puede llegar a converger prematuramente hacia un óptimo local. Es por lo anterior que actualmente existe la tendencia al uso de distribuciones mixtas como medio para capturar las no linealidades que se pudieran presentar en los problemas, lo cual es una poderosa herramienta utilizada por varias propuestas recientes.

Por otra parte, los esfuerzos de investigación actuales se han centrado en los algoritmos de estimación de distribución con el uso de cópulas como funciones de densidad conjuntas entre variables, esto gracias a la ventaja que presentan las cópulas de poder incorporar diferentes tipos de distribuciones para cada variable. Sin embargo, la mayoría de las propuestas utilizan sólo un tipo de cópula como función de distribución, lo cual limita el tipo de relaciones y la fuerza de estas relaciones que puede ser descrita. Además de que son pocas las propuestas que estiman los parámetros de las cópulas. Es hasta los trabajos más recientes donde se propone la estimación de la máxima verosimilitud y otros métodos estadísticos para encontrar las cópulas y parámetros ideales para cada relación entre variables. Sin embargo, todavía son pocos los trabajos que tienen estas características y falta incorporar el uso de algunos tipos de cópulas.

Debido a lo mencionado en los párrafos anteriores, es deseable buscar nuevas alternativas a las propuestas existentes para aprovechar los beneficios de los algoritmos de estimación de distribución, sin que esto signifique que el tipo de problemas que se pueden atacar sea limitado por las multimodalidades o alta dimensionalidad de éstos.

1.2 Planteamiento del problema

El problema general de optimización puede formularse de la siguiente manera:

Encontrar

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$

que optimice

$$f(\mathbf{x})$$

sujeto a

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, m$$

y

$$h_k(\mathbf{x}) = 0, \quad k = 1, \dots, p$$

Donde:

\mathbf{x} es el vector de variables de decisión pertenecientes a los números reales,

n es la dimensionalidad del problema,

$f(\cdot)$ es la función objetivo que se desea optimizar; ésto es, se desea encontrar un elemento \mathbf{x}_0 que minimice o maximice $f(\cdot)$,

$g_j(\cdot)$ son las restricciones de desigualdad a las que está sujeta la función objetivo,

m es el numero de restricciones de desigualdad,

$h_k(\cdot)$ son las restricciones de igualdad a las que esta sujeta la función objetivo, y

p es el numero de restricciones de igualdad.

Típicamente, en los algoritmos de estimación de distribución se realiza la optimización mediante la maximización de la función objetivo, sin embargo también se puede hacer la minimización de la función objetivo con un simple cambio de signo a la función objetivo:

$$\text{máx}(f(\mathbf{x})) = \text{mín}(-f(\mathbf{x}))$$

Las restricciones a las que está sujeta la función objetivo delimitan la(s) región(es) factible(s) dentro del espacio de búsqueda, donde se debe buscar la mejor solución.

Esta propuesta se concentrará en la resolución de problemas de alta dimensionalidad, con especial

énfasis en aquellos donde las relaciones entre las variables de decisión son fuertes, los cuales no han sido abordados con gran éxito por las propuestas del estado del arte de algoritmos de estimación de distribución.

Hipótesis. Es posible diseñar un algoritmo de estimación de distribución que haga uso de uno o varios modelos seleccionados automáticamente para la representación eficiente de las relaciones existentes en grupos disjuntos de variables, ésto a partir de los modelos y técnicas existentes en el estado del arte, y que permita obtener soluciones de buena calidad en problemas de alta dimensionalidad donde las relaciones entre variables son fuertes.

1.3 Objetivos generales y específicos del proyecto

1.3.1 Generales

- Contribuir al estado del arte de computación evolutiva con un nuevo algoritmo de estimación de distribución, cuyo énfasis de diseño será representar las interacciones entre las variables, mediante algún modelo gráfico o técnica de agrupación, y aprovechar esta representación para resolución eficiente de problemas con interacciones fuertes entre variables y alta dimensionalidad.

1.3.2 Específicos

- Obtener un estudio empírico sobre el desempeño de diferentes modelos probabilísticos en funciones escalables.
- Proponer un modelo cuyo énfasis será explotar información acerca de las interacciones entre variables.
- Obtener un estudio experimental acerca del uso de varios modelos simultáneamente, aplicados sobre grupos disjuntos de variables.

- Hacer una propuesta de un algoritmo de estimación de distribución, que emplee uno o varios modelos, así como un mecanismo para elección de éstos.

1.4 Organización de la tesis

El resto de la tesis está organizada de la manera siguiente. En el capítulo 2 se presentan los conceptos básicos acerca de algoritmos de estimación de distribución, en particular aquellos relacionados con el algoritmo propuesto, así como algunos trabajos presentes en el estado del arte de algoritmos de estimación de distribución para optimización numérica. En el capítulo 3 se detallan los componentes y el funcionamiento del algoritmo propuesto. En el capítulo 4 se detallan los experimentos llevados a cabo, así como los resultados obtenidos a partir de éstos. En el capítulo 5 se mencionan las conclusiones obtenidas en el desarrollo de este trabajo de tesis, así como ideas de trabajo futuro que podría llevarse a cabo para mejorar ésta propuesta o para el desarrollo de otras propuestas relacionadas.

2

Antecedentes

El uso de técnicas metaheurísticas para la resolución de problemas de optimización es muy frecuente debido a la gran cantidad de problemas que no pueden ser atacados eficientemente por técnicas de optimización clásica. La computación evolutiva ha sido, debido a los buenos resultados obtenidos, un punto de referencia entre estas técnicas. Durante la década pasada surgió un nuevo paradigma dentro de la computación evolutiva: los algoritmos de estimación de distribución. A continuación se presentan algunos conceptos en los que se basan los algoritmos de estimación de distribución, y algunos conceptos probabilísticos usados para describir la distribución de las variables y las relaciones entre éstas, además de algunas propuestas existentes en la literatura.

2.1 Algoritmos evolutivos

Es un conjunto de técnicas inspiradas en la evolución natural de las especies, donde una población de individuos compite en un ambiente de recursos limitados, lo que causa la selección de los individuos más aptos. Estos individuos actúan como padres de nuevos individuos los cuales son creados mediante la recombinación y la mutación. A través de la creación de nuevos individuos y de la selección natural

en cada generación, se llega a una mejora de la aptitud de la población en general.

Los técnicas principales dentro de los algoritmos evolutivos son:

- **Algoritmos genéticos.** Propuestos por Holland en 1975, pero popularizadas por Goldberg en 1989. Cada individuo de la población es representado mediante cromosomas, típicamente siendo estos cromosomas cadenas binarias. La selección natural se lleva a cabo al momento de seleccionar los padres de las nuevos individuos, a los que se les aplican operadores de cruce y mutación para producir una nueva generación de nuevos individuos.
- **Estrategias evolutivas.** Desarrolladas por Rechenberg y Schwefel en 1973. La representación de cada individuo es mediante números reales. Aquí se aplica un operador de reproducción para generar nuevos individuos, el cual puede ser de mutación y opcionalmente de cruce. La selección se puede llevar a cabo de dos formas, una donde se mezcla la población de padres e hijos y se selecciona a los mejores, y otra donde la población de hijos es mayor a la de padres y sólo se selecciona a los mejores hijos.
- **Programación evolutiva.** Propuesta por Fogel en 1962. Esta técnica inicialmente trata de evolucionar un algoritmo que actúa sobre un conjunto de símbolos de entrada de un alfabeto finito de tal manera que, a partir de una secuencia de símbolos, se maximice la salida del algoritmo. Posteriormente la propuesta se adaptó para su funcionamiento con variables reales. El único operador de variación es la mutación, la cual es parecida a la mutación en las estrategias evolutivas pero puede utilizar diferentes funciones de perturbación. La selección es mediante un tipo de torneo.
- **Programación genética.** Propuesta por Koza en los 90's. Intenta evolucionar programas computacionales. La representación es mediante árboles, los cuales son formados a partir de un conjunto de nodos terminales y un conjunto de nodos función con diferente aridad. La cruce se lleva a cabo mediante el intercambio de subárboles y la mutación por medio del reemplazo de un subárbol por un árbol generado aleatoriamente.

2.2 Marginales

La distribución marginal de un conjunto de variables aleatorias es la distribución de probabilidad de las variables contenidas en el conjunto. Dado que en este trabajo se consideran los marginales como univariados, la distribución de probabilidad se entenderá como la probabilidad de que una variable aleatoria tome cierto valor.

2.2.1 Distribución Beta

Es una distribución de probabilidad definida en el intervalo $(0, 1)$ que tiene dos parámetros de forma positivos, denotados típicamente por α y β .

2.2.1.1. Función de densidad de probabilidad

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1} (1-x)^{\beta-1}}{B(\alpha, \beta)}$$

donde $B(\alpha, \beta)$ es la función beta.

En la figura 2.1 se puede ver la función de densidad de probabilidad con diferentes parámetros.

2.2.1.2. Función de distribución acumulada

$$F(x; \alpha, \beta) = \frac{B_x(\alpha, \beta)}{B(\alpha, \beta)} = I_x(\alpha, \beta)$$

donde $B_x(\alpha, \beta)$ es la función beta incompleta e $I_x(\alpha, \beta)$ es la función beta incompleta regularizada.

En [Posten, 1993] se propone un algoritmo para evaluar la función de distribución beta.

El la figura 2.2 se puede ver la función de distribución acumulada con diferentes parámetros.

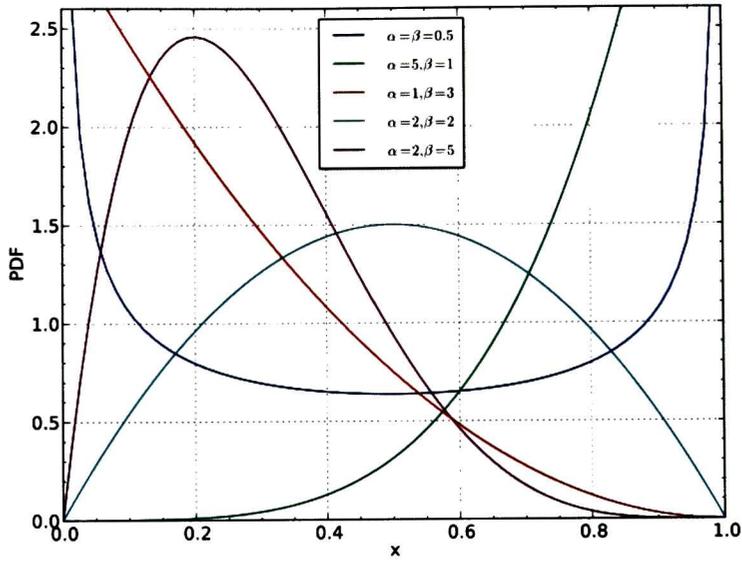


Figura 2.1: Función de densidad de probabilidad

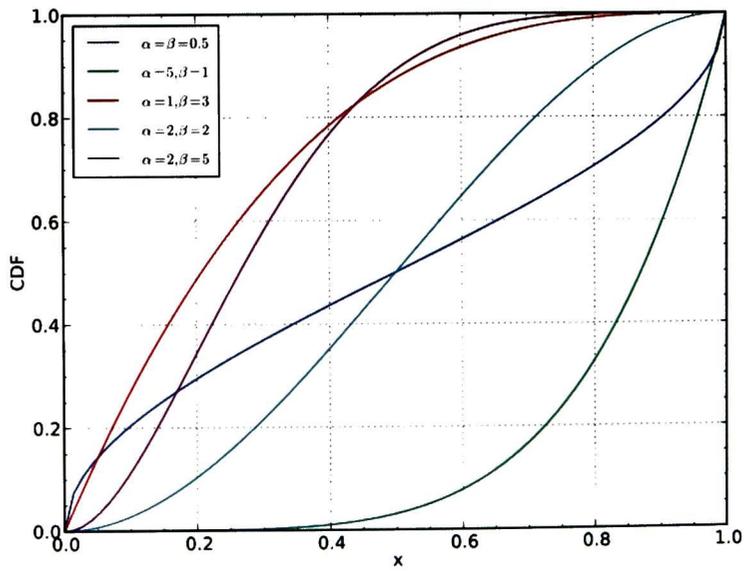


Figura 2.2: Función de distribución acumulada

2.2.1.3. Formas

La función beta puede tomar diferentes formas, como se muestra en la figura 2.1, dependiendo de los valores de los dos parámetros:

- $\alpha = 1, \beta = 1$ es la distribución uniforme $[0, 1]$
- $\alpha < 1, \beta < 1$ tiene forma de U (línea azul)
 - $\alpha = \frac{1}{2}, \beta = \frac{1}{2}$ es la distribución arco seno
- $\alpha < 1, \beta \geq 1$ o $\alpha = 1, \beta > 1$ es estrictamente decreciente (línea roja)
 - $\alpha = 1, \beta > 2$ es estrictamente convexa
 - $\alpha = 1, \beta = 2$ es una línea recta
 - $\alpha = 1, 1 < \beta < 2$ es estrictamente cóncava
- $\alpha = 1, \beta < 1$ o $\alpha > 1, \beta \leq 1$ es estrictamente creciente (línea verde)
 - $\alpha > 2, \beta = 1$ es estrictamente convexa
 - $\alpha = 2, \beta = 1$ es una línea recta
 - $1 < \alpha < 2, \beta = 1$ es estrictamente cóncava
- $\alpha > 1, \beta > 1$ es unimodal (líneas rosa y celeste)
- $\alpha = \beta$ la función densidad es simétrica (líneas azul y celeste)

Es esta variedad de formas que toma la distribución beta lo que la hace muy flexible y apta para representar distribuciones desconocidas, sin necesidad de recurrir a diferentes distribuciones y las correspondientes pruebas necesarias para seleccionar la mejor de estas.

2.2.1.4. Estimación de parámetros

Sea

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

la media de la muestra y

$$v = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

la varianza de la muestra. Las estimaciones de los parámetros por el método de momentos son:

$$\hat{\alpha} = \bar{x} \left(\frac{\bar{x}(1-\bar{x})}{v} - 1 \right) \quad (2.1)$$

$$\hat{\beta} = (1-\bar{x}) \left(\frac{\bar{x}(1-\bar{x})}{v} - 1 \right) \quad (2.2)$$

Cuando la distribución es requerida sobre un intervalo diferente a $[0, 1]$, digamos $[l, h]$, entonces reemplazamos \bar{x} con $\bar{x} = \frac{\bar{x} - l}{h - l}$, y v con $v = \frac{v}{(h - l)^2}$ en las ecuaciones 2.1 y 2.2.

No hay una forma cerrada de las estimaciones de máxima verosimilitud de los parámetros.

2.2.1.5. Generación de variables aleatorias

Para generar las variables se utiliza la función cuantil que es igual a la función beta incompleta inversa.

Una algoritmo para la estimación de esta función fue propuesto en [Cran et al., 1977]. Para más detalles de la distribución beta, puede referirse a [Johnson et al., 1995].

2.3 Cópulas

Definición 1. Para toda $d \geq 2$, una cópula de d dimensiones es una función de distribución de d variables en $[0, 1]^d$, donde los marginales univariados están uniformemente distribuidos en $[0, 1]$.

Esto es

$$C(u_1, \dots, u_d) = \Pr[U_1 \leq u_1, \dots, U_d \leq u_d]$$

donde $U_i \sim U(0, 1)$ para $i = 1, \dots, d$.

El siguiente teorema conocido como el teorema de Sklar [Sklar, 1959] es el fundamento de la teoría de cópulas.

Teorema 1. Sea F una función de distribución de d dimensiones con marginales univariadas F_1, F_2, \dots, F_d . Denote A_j el rango de F_j , $A_j = F_j(\mathbb{R})$ para $j = 1, 2, \dots, d$. Existe una cópula C tal que para todas $(x_1, x_2, \dots, x_d) \in \mathbb{R}^d$,

$$F(x_1, x_2, \dots, x_d) = C(F_1(x_1), F_2(x_2), \dots, F_d(x_d)) \quad (2.3)$$

Tal C es determinada únicamente en $A_1 \times A_2 \times \dots \times A_d$, y por tanto, es única cuando F_1, F_2, \dots, F_d son todos continuos.

Usando la ecuación 2.3, cualquier cópula puede ser combinada con diferentes funciones de distribución univariadas para obtener una función de distribución multivariable.

Para más información sobre teoría de cópulas puede referirse a [Durante and Sempi, 2010, Nelsen, 1998, Cherubini et al., 2004].

2.3.1 Cópulas arquimedeanas

Hablamos de cópulas arquimedeanas cuando éstas pueden construirse usando una función $\varphi : I \rightarrow \mathbb{R}^{*+}$, continua, decreciente, convexa y tal que $\varphi(1) = 0$. Tal función es llamada generador. Es llamado un generador estricto cuando $\varphi(0) = +\infty$.

La pseudo-inversa de φ debe ser definida como sigue:

$$\varphi^{-1}(v) = \begin{cases} \varphi^{-1}(v) & 0 \leq v \leq \varphi(0) \\ 0 & \varphi(0) \leq v \leq +\infty \end{cases}$$

Esta pseudo-inversa es tal que, por composición con el generador, da la identidad, como lo hacen las inversas ordinarias para las funciones con dominio y rango \mathfrak{R} :

$$\varphi^{-1}(\varphi(v)) = v \quad \text{para cada } v \in I$$

Además, coincide con la inversa usual si φ es un generador estricto.

Teorema 2. Sea φ un generador. La función $C : [0, 1]^n \rightarrow [0, 1]$ definida por

$$C(u_1, \dots, u_d) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_n))$$

es una cópula si y solo si φ^{-1} es completamente monotónico en $[0, \infty]$.

Definición 2. Sea φ un generador estricto, con φ^{-1} completamente monotónico en $[0, \infty]$. Entonces, una cópula arquimedea de n variables es la función

$$C(u_1, \dots, u_d) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_n))$$

La cópulas arquimedeanas son simétricas, en el sentido que

$$C(v, z) = C(z, v)$$

También son asociativas

$$C(C(v, z), u) = C(v, C(z, u)) \quad \text{para toda } (v, z, u) \in I^3$$

dado que ambos lados de la igualdad anterior se reducen a $\varphi^{-1}(\varphi(v) + \varphi(z) + \varphi(u))$.

A continuación se enumeran algunas cópulas arquimedeanas relevantes.

2.3.1.1. Cópula de Frank

El generador esta dado por

$$\varphi(u) = -\ln\left(\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right)$$

por lo tanto

$$\varphi^{-1}(t) = -\frac{1}{\theta} \ln(1 + e^{-t}(e^{-\theta} - 1))$$

es completamente monótonico si $\theta > 0$. La n -cópula de Frank esta dada por:

$$C(u_1, \dots, u_n) = -\frac{1}{\theta} \ln \left\{ 1 + \frac{\prod_{i=1}^n (e^{-\theta u_i} - 1)}{(e^{-\theta} - 1)^{n-1}} \right\} \text{ con } \theta > 0 \text{ cuando } n \geq 2$$

La cópula de Frank es simétrica, puesto que muestra igual dependencia en ambas colas. Se puede ver esto en las figuras 2.3a y 2.3b.

2.3.1.2. Cópula de Gumbel

El generador esta dado por $\varphi(u) = (-\ln(u))^\theta$, por lo tanto $\varphi^{-1}(t) = \exp\left(-t^{\frac{1}{\theta}}\right)$; es completamente monótonico si $\theta > 1$. Por lo tanto, la n -cópula de Gumbel es:

$$C(u_1, \dots, u_n) = \exp \left\{ - \left[\sum_i (-\ln u_i)^\theta \right]^{\frac{1}{\theta}} \right\} \text{ con } \theta > 1$$

La cópula de Gumbel, también llamada cópula de Gumbel-Hougar, es una cópula asimétrica que muestra mayor dependencia en la cola positiva que en la cola negativa. Esto se puede ver en las figuras 2.4a y 2.4b.

2.3.1.3. Cópula de Clayton

El generador esta dado por $\varphi(u) = \frac{(u^{-\theta} - 1)}{\theta}$, por lo tanto $\varphi^{-1}(t) = (1 + \theta t)^{-\frac{1}{\theta}}$; es completamente monótonico si $\theta > 0$. Por lo tanto, la n -cópula es:

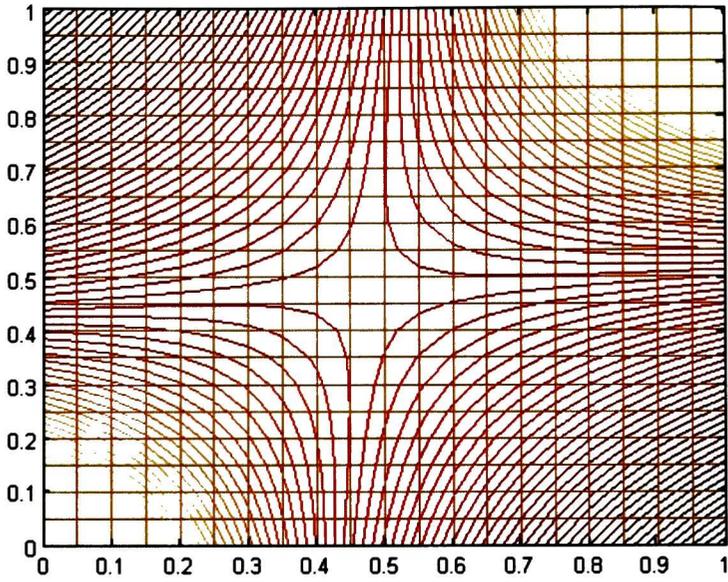
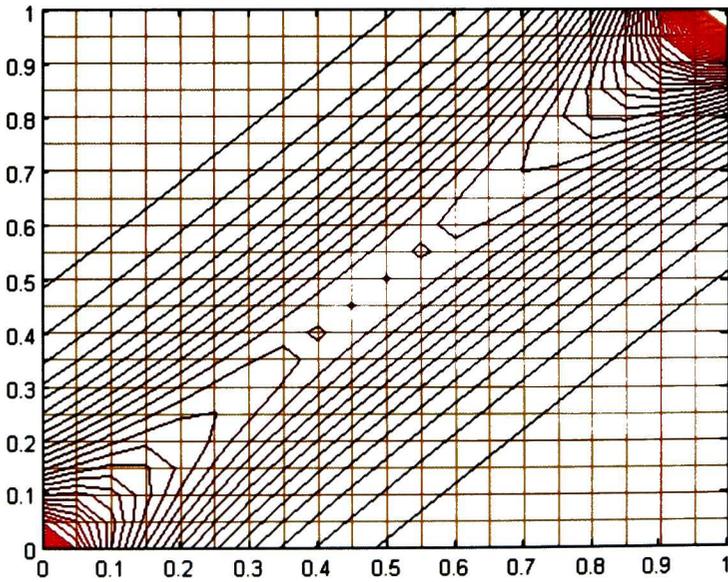
(a) $\theta = 0.5$ (b) $\theta = 8$

Figura 2.3: Cópula de Frank

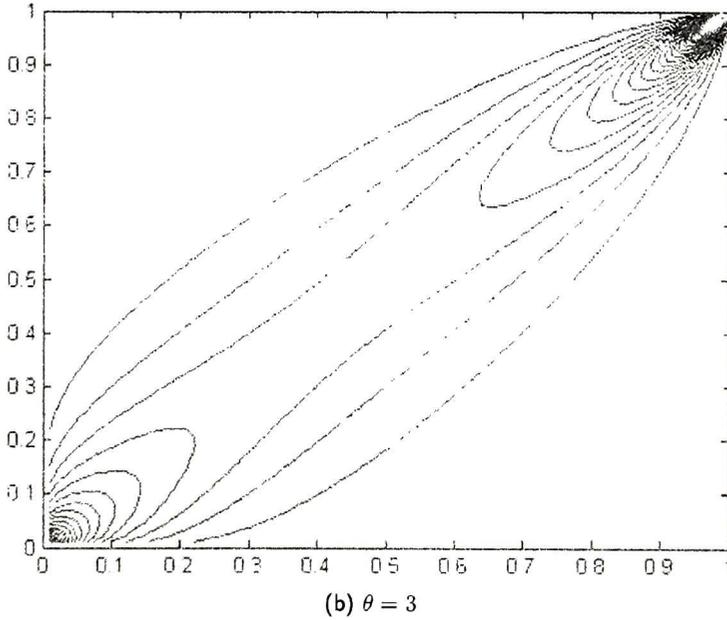
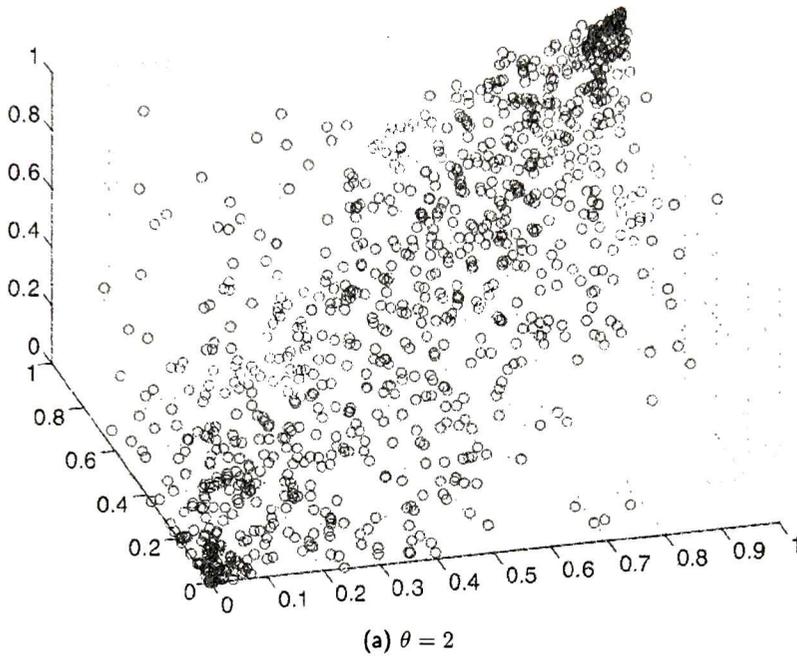


Figura 2.4: Cópula de Gumbel

$$C(u_1, \dots, u_n) = \left[\sum_i u_i^{-\theta} - n + 1 \right]^{-\frac{1}{\theta}} \quad \text{con } \theta > 0$$

La cópula de Clayton es una cópula asimétrica que muestra mayor dependencia en la cola negativa que en la cola positiva. Esto se puede ver en las figuras 2.5a y 2.5.

2.4 Modelos gráficos probabilísticos

La representación de un modelo gráfico probabilístico consiste de dos componentes:

- Estructura: es un grafo acíclico dirigido que representa un conjunto de (in)dependencias condicionales en las variables de \mathbf{X} .
- Conjunto de distribuciones de probabilidad generalizadas locales. Esto es, las probabilidades de cada variable dadas las probabilidades de las variables con las que esta directamente relacionada.

2.4.1 Redes bayesianas

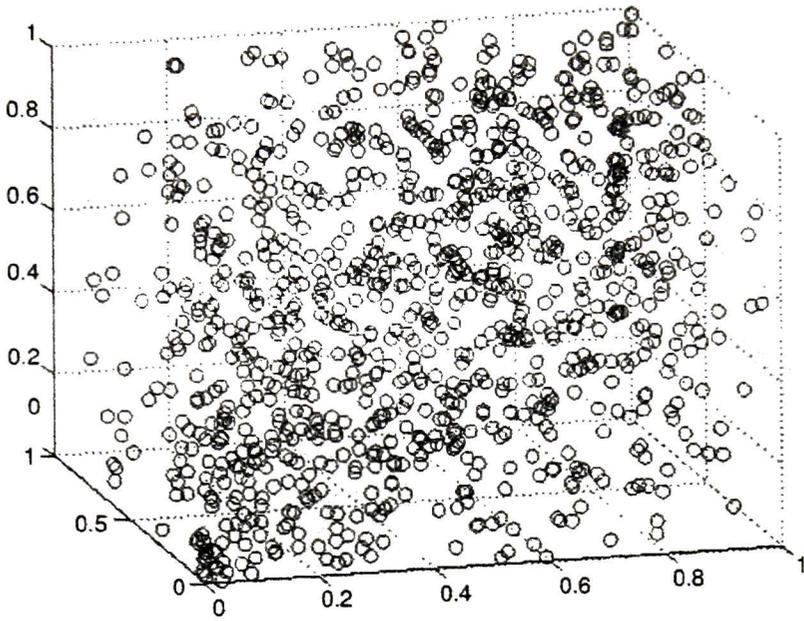
Este tipo de modelo probabilístico se utiliza en el caso de que cada variable $X_i \in \mathbf{X}$ sea discreta.

Si la variable X_i tiene r_i valores posibles, $x_i^1, \dots, x_i^{r_i}$, la distribución local, $p(x_i^k | pa_i^{j,S}, \theta_i)$ es una distribución discreta sin restricciones:

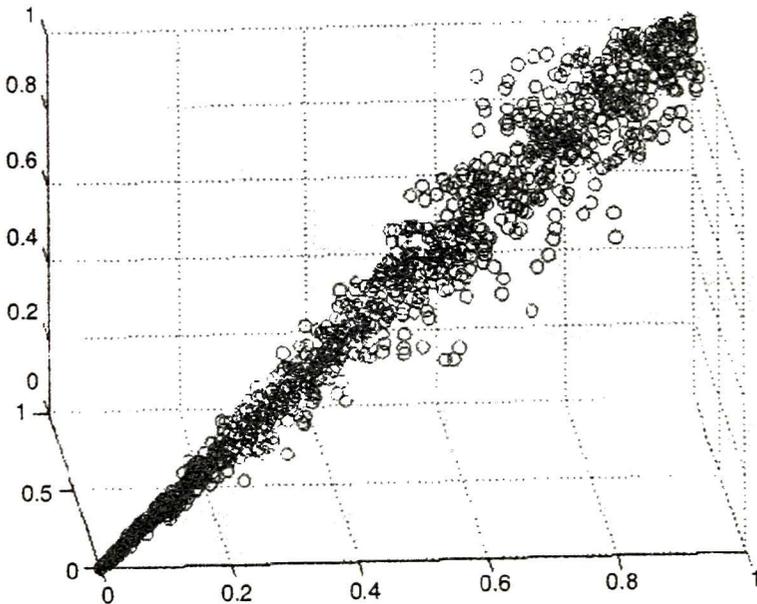
$$p(x_i^k | pa_i^{j,S}, \theta_i) = \theta_{x_i^k | pa_i^j} \equiv \theta_{ijk} \quad (2.4)$$

donde $pa_i^{j,S}, \dots, pa_i^{q_i,S}$ denota los valores de Pa_i^S , el conjunto de padres de la variable X_i en la estructura. El término q_i denota el número de instancias diferentes posibles de los padres de la variable X_i . Así, $q_i \prod_{X_g \in Pa_i^S} r_g$.

Los parámetros locales son dados por $\theta_i = ((\theta_{ijk})_{k=1}^{r_i})_{j=1}^{q_i}$. En otras palabras, el parámetro θ_{ijk} representa la probabilidad condicional de la variable X_i tenga el valor j , sabiendo que el conjunto de sus padres tienen el valor k .



(a) $\theta = 0.53$



(b) $\theta = 15$

Figura 2.5: Cópula de Clayton

2.4.2 Redes Gaussianas

Este tipo de modelo probabilístico se utiliza en el caso de que cada variable $X_i \in \mathbf{X}$ sea continua y cada función local de densidad es el modelo de regresión lineal:

$$p(x_i | pa_i^S, \theta_i) \equiv \mathcal{N} \left(x_i; m_i + \sum_{x_j \in pa_i} b_{ji} (x_j - m_j), v_i \right) \quad (2.5)$$

donde $\mathcal{N}(x; \mu, \sigma^2)$ es una distribución normal univariable con media μ y varianza σ^2 . Dada esta forma un arco faltante de X_j a X_i implica que $b_{ji} = 0$ en el modelo de regresión lineal anterior. Los parámetros locales son dados por $\theta_i = (m_i, \mathbf{b}_i, v_i)$, donde $\mathbf{b}_i = (b_1, \dots, b_{i-1})^t$ es un vector columna.

La interpretación de los componentes de los parámetros locales es como sigue: m_i es la media incondicional de X_i , v_i es la varianza condicional de X_i dado \mathbf{Pa}_i , y b_{ji} es un coeficiente lineal que refleja la fuerza de la relación entre X_j y X_i .

2.4.3 Vines

Una *vine* de n variables es un conjunto de árboles anidados, donde las aristas del árbol j son los nodos del árbol $j + 1$, para $j = 1, \dots, n - 2$, y cada árbol tiene el máximo número de aristas. Los *vines*, dado que son grafos y existen distintos tipos de *vines* donde algunos no contienen ciclos, recientemente han sido utilizados como el componente estructura de modelos gráficos probabilísticos [Kurowicka and Cooke, 2002, Salinas-Gutiérrez et al., 2010].

2.4.4 Métodos de inducción

Los métodos de inducción de los modelos gráficos probabilísticos tratan aprender o construir automáticamente la estructura o grafo del modelo.

2.4.4.1. *Métodos de exclusión de aristas.*

En este tipo de métodos se tiene inicialmente un grafo con la mayoría o todas las relaciones que se pueden dar entre las variables y se trata de eliminar aristas de este grafo inicial detectando independencias entre variables.

2.4.4.2. *Métodos score+search.*

Hacen una búsqueda en el espacio de estructuras de redes Gaussiana mediante heurísticas como la búsqueda voraz, recocido simulado, búsqueda tabú, algoritmos genéticos, etc.; utilizando medidas de las estructuras candidatas como función objetivo.

2.4.5 Estimación de la máxima verosimilitud.

Sea $\mathbf{Y} = (Y_1, \dots, Y_n)^T$ un vector de datos y sea $\theta = (\theta_1, \dots, \theta_n)^T$ un vector de parámetros. Sea $f(\mathbf{Y} | \theta)$ la densidad de \mathbf{Y} , que depende de los parámetros.

La función $L(\theta) = f(\mathbf{Y} | \theta)$ vista como una función de θ con \mathbf{Y} fija en los datos observados es llamada la función de verosimilitud. La estimación de la máxima verosimilitud es el proceso de estimar el valor de θ que maximiza la función de verosimilitud.

2.5 Algoritmos de estimación de distribución.

Los algoritmos de estimación de distribución son técnicas de optimización que se basan en la inducción y el muestreo de modelos probabilísticos para la exploración del espacio de búsqueda. El objetivo de este tipo de técnicas es tomar en cuenta las dependencias entre las variables de los mejores individuos de una población, en lo que se podría decir, un proceso de aprendizaje del problema, y transmitirla a la siguiente generación.

2.5.1 Funcionamiento

Los algoritmos de estimación de distribución trabajan con una población de individuos inicialmente generada mediante una distribución uniforme en el rango permitido por cada variable de decisión. La población se evalúa mediante una función objetivo a partir de la cual se le puede asignar una aptitud para poder medir la calidad del individuo. A partir de la aptitud asignada a cada solución se puede realizar una selección de las mejores soluciones mediante un operador de selección específico. Una vez seleccionado un conjunto de mejores soluciones, se crea o ajusta un modelo probabilístico para intentar aproximar la distribución probabilística de los individuos seleccionados. Después se muestrean nuevas soluciones a partir del modelo probabilístico, las cuales formarán parte de una nueva generación en el algoritmo, ya sea reemplazando algunas o todas las soluciones de la población anterior. El algoritmo continúa este proceso iterativamente hasta que se cumple cierta condición de paro. El pseudo-código general de un algoritmo de estimación de distribución es el siguiente.

Empezar

 Inicializar población con soluciones candidatas aleatorias;

 Evaluar cada candidato y asignar aptitud;

 Repetir

 Seleccionar mejores soluciones candidatas;

 Inducir modelo probabilístico a partir de las mejores
 soluciones candidatas;

 Generar nueva población a partir del modelo probabilístico;

 Evaluar nuevos candidatos;

 Hasta (condición de termino sea cumplida)

Finalizar

2.5.2 EDAs en optimización numérica

En esta subsección se presentan algunos de los trabajos encontrados en la literatura referentes a la optimización numérica mediante algoritmos de estimación de distribución, organizados de acuerdo con el enfoque que utilizan para representar el espacio de búsqueda y la complejidad de los modelos utilizados para representar las relaciones entre variables.

2.5.2.1. Discretización

En [Tsutsui et al., 2001], se propone dividir el espacio de búsqueda de cada variable de decisión mediante el uso de histogramas. El histograma de cada variable estará dividido en H intervalos, siendo H un parámetro. Se usan dos tipos de histogramas: de anchura fija y de altura fija. En los histogramas de anchura fija cada intervalo tiene la misma anchura y la probabilidad de cada intervalo se da según la densidad de individuos en cada intervalo. En los histogramas de altura fija cada intervalo tiene la misma densidad de individuos y altura y por lo tanto la misma probabilidad; las áreas con mayor densidad de individuos tendrán mayor número de intervalos, los cuales tendrán menor anchura, y en consecuencia se elevará la precisión. La generación de nuevos individuos es un muestreo con distribución uniforme en el intervalo escogido a través del método de ruleta rusa o del método de muestreo estocástico universal. Ambos tipos de histogramas muestran ser eficientes en funciones donde no hay o es débil la relación entre las variables, además de ser capaces de capturar multimodalidades; en cambio, en funciones con un mediano nivel de interacciones entre variables no funciona bien.

En [Ding et al., 2006], se propone el uso de histogramas de anchura fija donde la probabilidad de cada intervalo se actualiza utilizando la probabilidad del intervalo en la generación pasada y la probabilidad en la generación actual; se demuestra que esta propuesta mejora la propuesta de histogramas de anchura fija en [Tsutsui et al., 2001]. En [Ding et al., 2008], se mejora el rendimiento de la propuesta en [Ding et al., 2006] mediante el uso de dos estrategias: influenciar la probabilidad de un intervalo con la probabilidad de los intervalos vecinos, lo que permite reducir la población necesaria para estimar el modelo probabilístico, y dividir un intervalo si la densidad de individuos

supera un umbral, lo que permite encontrar soluciones con mayor precisión.

En [Chen et al., 2006], se propone un método de discretización a demanda el cual divide en dos cada región del espacio de búsqueda de cada variable si contiene un número mayor a $N \times \gamma$ individuos, donde N es el tamaño de la población y γ es un parámetro que se ajusta en cada iteración del algoritmo con la fórmula $\gamma_{i+1} = \gamma_i \times \epsilon$, con $0 < \gamma < 1$ y $0 < \epsilon < 1$. Este método de discretización se utiliza para crear codificaciones que serán usadas por modelos para optimización combinatoria, como el propuesto en [Harik et al., 2006]. La ventaja de este enfoque es el gran desarrollo que se tiene en los algoritmos de estimación de distribución para optimización combinatoria, incluyendo los que toman en cuenta las relaciones entre variables.

2.5.2.2. Representación directa

Sin dependencias

En este tipo de enfoques cada variable se supone independiente de las otras y la función de densidad conjunta del vector $\mathbf{x} = (x_1, \dots, x_n)^T$ de variables de decisión puede ser factorizada de acuerdo con $p(\mathbf{x}) = \prod_{i=1}^n p(x_i)$. Por lo general, la distribución utilizada es la normal, lo que hace que solo se puedan capturar distribuciones donde las mejores soluciones están situadas alrededor de un sólo punto en el espacio de búsqueda.

En [Rudlof and Köppen, 1997], se mantiene un vector con la media de cada variable y la desviación estándar es global. En cada generación la media se mueve hacia la mejor solución y la desviación estándar, la cual es asignada inicialmente con un valor grande para garantizar la exploración del espacio de búsqueda, se reduce por un factor multiplicativo constante para concentrar sucesivamente la búsqueda en una región más pequeña. Los resultados muestran buen desempeño en las funciones de dos picos y Rastrigin. La desventaja de esta propuesta es asumir que todas las variables tienen la misma desviación estándar.

En [Sebag and Ducoulombier, 1998], se propone una extensión del algoritmo propuesto en [Baluja, 1994]. Cada variable tiene una media y desviación estándar propias, lo que hace que se desempeñe bien en escenarios donde algunas variables tienen mayor varianza que otras. La adaptación de la media de cada variable se realiza mediante la combinación de los dos mejores individuos y el

peor: $\mu_i^{t+1} = (1 - \alpha) \cdot \mu_i^t + \alpha \cdot (x_{ibest,1} + x_{ibest,2} - x_{iworst})$, donde α es una constante. Se proponen 4 estrategias para la actualización de la desviación estándar: usar un valor constante, usar una estrategia evolutiva $(1, \lambda)$ para ajustarla, calcular la varianza de los K mejores individuos de cada generación o mediante uso de una regla parecida a la que se usa para actualizar la media. Esta propuesta se comporta mejor que estrategias las evolutivas estándar y el PBIL booleano [Baluja, 1994], este último aplicado a los problemas en su forma discreta.

En [Servet et al., 1998] se propone otra variación de [Baluja, 1994], donde cada variable es modelada mediante un intervalo y la probabilidad de que la variable tome un valor mayor a la mitad del intervalo, si la probabilidad es menor a 0.1 el intervalo se recorta a la mitad menor, y si la probabilidad es mayor a 0.9 el intervalo se recorta a la mitad mayor.

Los trabajos [Larrañaga et al., 1999, Larrañaga et al., 2000] proponen UMDA_c, el cual en cada generación y por cada variable lleva a cabo pruebas estadísticas para encontrar la función de densidad que mejor concuerde con cada variable y después se estiman los parámetros de cada función de densidad mediante la estimación de la máxima verosimilitud.

Dependencias de dos variables

En [Larrañaga et al., 1999, Larrañaga et al., 2000] se propone MIMIC_c^G, una adaptación de [Bonet et al., 1997] para dominios continuos. El modelo probabilístico para cada par de variables es una gaussiana de dos variables. El algoritmo pretende aprender las relaciones entre pares de variables con un algoritmo voraz parecido al de [Bonet et al., 1997], que escoge primero la variable Y con la menor varianza y después entre las demás variables busca la variable X que minimice $\frac{\sigma_X^2 \sigma_Y^2 - \sigma_{XY}^2}{\sigma_Y^2}$ y son vinculadas estas dos variables.

En [Wang and Zeng, 2010, Wang et al., 2009b] se propone un algoritmo que utiliza cópulas como modelo para representar la distribución conjunta. Una cópula gaussiana se construye mediante un único parámetro estimado con el método de la máxima verosimilitud. Para generar nuevos individuos se utiliza el método de Mote Carlo. En [Wang et al., 2009a, Wang and Zeng, 2010], se propone el uso de las copulas arquimedeanas Clayton y Ali-Mikhail-Haq, y el método de distribución condicional como forma de generar dos variables a partir de la cuales se usara la quasi-inversa

de la cópula para generar los nuevos individuos. Estas propuestas muestran mejores resultados que [Sebag and Ducoulombier, 1998], además las propuestas que utilizan cópulas arquimedeanas muestran mejor desempeño que la propuesta que utiliza la cópula gaussiana.

En [Wang et al., 2010a] se propone el uso de distribuciones marginales empíricas para modelar la distribución de cada variable y para la distribución conjunta se utilizan cópulas de Gumbel con parámetros fijos. En [Wang et al., 2010b], se propone un algoritmo parecido al anterior pero se utilizan cópulas de Clayton para describir la distribución conjunta. El desempeño de la propuesta en [Wang et al., 2010a] es mejor que el de [Wang et al., 2010b], sin embargo el desempeño de [Wang et al., 2010a] es comparable con el de las propuestas en [Larrañaga et al., 2000]. Esto se debe a que los parámetros de las cópulas son fijos y su elección es un factor importante en la descripción de las relaciones entre variables.

En [Cuesta-Infante et al., 2010] se propone el uso de cópulas empíricas. Las copulas $C_{i-1,i}(u_{i-1}, v_i)$ se forman a partir de la variable x_i y una nueva variable y_i , donde la variable $y_i = C_{i-1,i}(G_{i-1}(y_{i-1}), F_i(x_i))$ con $i = 1, \dots, n - 1$; siendo G_j la función de distribución acumulativa (FDA) de y_j y F_j la FDA de x_j , para $j = 1, \dots, n$. En los experimentos llevados a cabo el algoritmo propuesto es claramente superado por un algoritmo que utiliza una distribución gaussiana multi-variable, sin embargo cuando se aumenta el número de variables el desempeño puede ser comparado al de UMDA_c [Larrañaga et al., 1999, Larrañaga et al., 2000].

En [Salinas-Gutiérrez et al., 2011] se propone utilizar un árbol de dependencias para expresar gráficamente las dependencias de las variables y modelar las relaciones mediante funciones de copulas de dos variables. El árbol se forma mediante el algoritmo de Kruskal para encontrar el árbol recubridor mínimo donde cada nodo tiene, a lo más, un nodo padre, lo que formaría un tipo de cadena. El árbol óptimo es el que produce mayor información entre los pares de variables. Se utiliza la máxima función de verosimilitud para escoger entre un conjunto de funciones de cópula y además estimar los parámetros de la cópula. Se demuestra que en la selección de cópulas comparada con la utilización de una cópula gaussiana es mejor el desempeño de la propuesta con selección de cópulas en las funciones Schwefel, Two Axes y Zakharov, mientras en otras es comparable con el uso de una cópula gaussiana.

Dependencias de múltiples variables

En [Larrañaga et al., 1999, Larrañaga et al., 2000] se proponen los algoritmos $EGNA_{ee}$ y $EGNA_{BGe}$, ambos utilizan redes gaussianas como modelos probabilísticos. En el primero se utiliza un método de exclusión de aristas y en el segundo se utiliza un método score+search para aprender la estructura de la red gaussiana. Los parámetros de la red gaussiana son aprendidos mediante la estimación de la máxima verosimilitud. Una desventaja de esta propuesta es que debido al uso de gaussianas simples no se pueden describir no linealidades. Además, el cálculo de los coeficientes de la red gaussiana se vuelve cada vez mas difícil conforme aumenta la dimensionalidad del problema.

En [Bosman and Thierens, 2000b] se propone un framework que puede ser utilizado en optimización continua y discreta. Se utiliza una distribución truncada por el peor individuo de la generación pasada y se propone reemplazar solo una parte de la población, de una manera parecida a una estrategia evolutiva.

En [Ocenasek et al., 2004] se propone un algoritmo que puede tratar con variables discretas y continuas. La distribución de cada variable se modela mediante un árbol de dependencias, donde cada variable de la que depende la variable actual sirve para dividir el árbol. Las variables continuas deben ser discretizadas para poder aplicar métricas bayesianas y saber en qué grado influyen en la otra variable. Cada hoja del árbol tiene una distribución normal.

En [Wang and Zeng, 2010] se propone el uso de cópulas empíricas para modelar las relaciones entre variables y se utiliza el método propuesto en [Strelen and Nassaj, 2007] para construir la cópula y muestrear nuevos individuos a partir de ésta. La desventaja de esta propuesta es que depende del parámetro de selección y de la estrategia con la que se calcula la varianza usada para muestrear individuos.

En [Cuesta-Infante et al., 2010] se propone el uso de cópulas arquimedeanas fijadas *a priori* y un método iterativo para muestrear nuevos individuos donde se obtienen los valores de las variables en cada iteración a partir de las variables con las cuales se tiene una relación y de las que ya se tiene un valor obtenido en iteraciones pasadas. Una ventaja de este trabajo es que se propone formas cerradas de la inversa de la cópula, lo que hace mas rápido el proceso de muestreo de nuevos individuos. Una desventaja es que los parámetros de las cópulas son fijos.

En [Salinas-Gutiérrez et al., 2010] se propone un algoritmo que utiliza un *vine* regular como modelo gráfico y cópulas gaussianas como las funciones para modelar las dependencias entre las variables. Se utiliza un algoritmo voraz para formar una cadena; primero se elige una permutación de tres variables, donde su cópula tenga la menor entropía, y se forman de manera aleatoria. Después se van añadiendo las variables que maximizan la información mutua respecto a las variables en los extremos, evitando hacer una cadena circular. A partir de esta permutación se construye un *vine* regular y se forman las cópulas. Se utilizan distribuciones beta para describir cada variable y los parámetros de las cópulas se estiman mediante la máxima verosimilitud. La desventaja de esta propuesta es que el *vine* puede llegar a ser una estructura muy compleja para ciertos problemas donde las relaciones son mas simples o no existen, lo que hace que el algoritmo pueda ser comparable en desempeño con algoritmos que utilizan modelos mas simples como $UMDA_c$.

En [Guo et al., 2011] se propone utilizar marginales empíricos y estimar los parámetros de una cópula arquimedean, ya sea la de Gumbel o la de Clayton, mediante la estimación de la máxima verosimilitud. En los experimentos realizados la cópula de Clayton presenta mejores resultados que la cópula de Gumbel, sin embargo, el uso de cualquiera de las dos cópulas muestran mejores resultados que $UMDA_c^G$ y $MIMIC_c^G$ [Larrañaga et al., 1999, Larrañaga et al., 2000].

Distribuciones mixtas

En [Gallagher, 2000] se propone un modelo de densidades mixtas llamado *AMix* donde cada variable es representada por la suma de varios componentes, cada uno con distribución gaussiana, y se utiliza un coeficiente para asignar el grado de contribución de cada componente. El número de componentes que tiene cada variable puede cambiar en cada iteración; la distancia Mahalanobis es utilizada para decidir si se crea un nuevo componente. El modelo se actualiza en cada iteración utilizando un conjunto de reglas. En este trabajo también se propone una extensión al algoritmo propuesto en [Baluja, 1994] continuo llamada *Fink*, donde cada variable se modela mediante un numero fijo de kernels gaussianos. Para llevar a cabo la actualización de los parámetros se toma en cuenta el mejor individuo de cada kernel y el mejor individuo global, esto para realizar una búsqueda cooperativa, además de que este método es más rápido que el utilizado en *AMix*. La contribución

de cada kernel gaussiano es equitativa.

En [Bosman and Thierens, 2000a] se propone agrupar las soluciones mediante los métodos *k-means* o *leader* y después crear una red gaussiana para cada grupo. El modelo gráfico se construye mediante algoritmos voraces incrementales, utilizando tests de hipótesis y penalizaciones por complejidad. También se propone la utilización de distribuciones gaussianas mixtas para capturar las no linealidades que se pudieran presentar en cada grupo, para lo cual se utiliza el algoritmo EM como método de estimación. La desventaja de esta propuesta es que no se comparte información entre los modelos probabilísticos de cada grupo.

En [Ahn et al., 2006] se plantea una extensión de [Pelikan et al., 1999] al dominio continuo, donde se usan redes bayesianas para representar las dependencias entre variables y se utilizan distribuciones gaussianas mixtas para representar cada variable. El modelo gráfico se crea mediante un algoritmo voraz que en cada iteración añade la arista que maximiza la medida llamada *Bayesian Information Criterion*. Una ventaja de esta propuesta es que mediante la red bayesiana se hace una descomposición del problema en cuestión, lo que permite se calculen los parámetros de las factorizaciones de una forma mas rápida. Se demuestra que esta propuesta supera a las propuestas en [Larrañaga et al., 2000, Bosman, 2003, Ocenasek and Schwarz, 2002].

Algoritmos adaptativos

En [Santana et al., 2008] se propone un algoritmo donde en cada iteración se debe aprender el grafo de relaciones entre variables, a partir del cual se seleccionará una clase de los modelos gráficos o una estrategia de aproximación junto con alguno de los métodos de inferencia de cada clase, para aproximar la distribución de las variables y sus (in)dependencias. Se demuestra que esta propuesta supera a los algoritmos con un modelo probabilístico estático.

En [Lobo and Lima, 2010] se propone una hiper-heurística basada en las ideas propuestas en [Harik and Lobo, 1999]. La hiper-heurística propone seleccionar entre un número finito de algoritmos de estimación de distribución sin dependencias, con dependencias de dos variables y dependencias de múltiples variables. Cada algoritmo se correrá por un tiempo relacionado con la complejidad del modelo utilizado para representar las relaciones entre variables, donde los algoritmos con modelos

menos complejos correrán más tiempo que los algoritmos que tienen modelos más complejos. Un algoritmo será descartado si la media de su población es menor a la media de la población de un algoritmo con mayor complejidad.

2.6 Resumen

En este capítulo se presentaron los algoritmos evolutivos, los cuales son técnicas inspiradas en la evolución natural de las especies y sus principales técnicas son: los algoritmos genéticos, las estrategias evolutivas, la programación evolutiva y la programación genética.

También se definieron las distribuciones marginales, que para el caso de este trabajo se entenderán como la probabilidad de que una variable aleatoria tome cierto valor. En específico se presentó la distribución beta, la cual está definida en el intervalo $(0, 1)$ y tiene dos parámetros de forma positivos: α y β los cuales pueden ser estimados a partir de la media y la varianza de una muestra. La distribución beta puede adoptar varias formas según los valores que tomen sus parámetros.

Además, se pudo dar una breve introducción a la teoría de cópulas, las cuales son funciones de distribución de 2 o más variables, donde cada variable está representada mediante una función de distribución. Las cópulas arquimedeanas son una familia de cópulas que tienen la forma $C(u_1, \dots, u_d) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_n))$ y tienen la propiedad de ser asociativas y simétricas. Dentro de esta familia se presentaron las cópulas de Frank, Gumbel y Clayton las cuales describen dependencia simétrica, asimétrica con mayor dependencia en la cola positiva y asimétrica con mayor dependencia en la cola negativa, respectivamente.

Se pudo ver que los modelos probabilísticos consisten de dos componentes: la estructura o grafo que representa las (in)dependencias entre variables, y el conjunto de probabilidades de cada variable dadas las probabilidades de las variables con las que está relacionada. Las redes bayesianas son un tipo de modelo gráfico probabilístico donde las variables son discretas. Cuando las variables son continuas y tienen distribución gaussiana las redes gaussianas son el modelo probabilístico gráfico más común. Los *vines* son grafos en forma de árboles anidados que recientemente se han utilizado como el componente estructura en algunos modelos gráficos probabilísticos. Se mencionaron además

dos tipos de métodos para inducir la estructura de un modelo gráfico probabilístico, uno que consiste en eliminar las aristas de un grafo detectando las independencias, y otro que consiste en la búsqueda de un grafo mediante heurísticas y la utilización de medidas de las estructuras candidatas. Como método de estimación de las probabilidades de cada variable, se menciona el método de la máxima verosimilitud, el cual estima los parámetros de las funciones de probabilidad dando mayor probabilidad al conjunto de observaciones que se utiliza como entrada del método.

Por último, se dieron a conocer los algoritmos de estimación de distribución, los cuales están basados en los algoritmos evolutivos con la diferencia de que utilizan un modelo probabilístico para explorar el espacio de búsqueda. En el estado del arte de los algoritmos de estimación de distribución en optimización numérica se dieron a conocer los diferentes enfoques existentes organizados de acuerdo a la representación del espacio de búsqueda (discreta o directa) que utilizan y la complejidad de los modelos probabilísticos utilizados, que van desde no tomar en cuenta las relaciones hasta modelos con dependencias multivariables.

3

Enfoque propuesto

A continuación se presentan los bloques de los cuales se compone el EDA propuesto en este trabajo. La propuesta se centra en la utilización de una red bayesiana de cópulas como modelo gráfico probabilístico, el cual fue recientemente propuesto y aun no utilizado dentro de los algoritmos de estimación de distribución. Como densidades locales de la red bayesiana de cópulas se utilizan 3 cópulas de la familia arquimedea, las cópulas de Frank, Gumbel y Clayton, las cuales describen cada una un tipo diferente de dependencia. La utilización de más de una cópula por un algoritmo de estimación de distribución sólo se ha reportado en un trabajo, sin embargo tal trabajo se limita a utilizar cópulas de 2 variables, a diferencia de las cópulas de 2 o más variables utilizadas aquí. La estructura de la red bayesiana es aprendida mediante una maquinaria estándar que consiste en un algoritmo voraz que prueba cada una de las aristas posibles y agrega la que signifique un incremento mayor a la métrica BIC. Esta maquinaria ha sido modificada para que en lugar de calcular la verosimilitud por cada arista que se prueba (lo que sería cada vez más costoso computacionalmente conforme aumenta la dimensionalidad del problema) se utilice el tau de Kendall como medida de relación entre variables. Además de esto, el parámetro de las cópulas puede ser estimado directamente a partir del tau de Kendall, en vez de utilizar el método de la máxima verosimilitud, lo que significa

menor costo computacional.

3.1 Selección de individuos

La selección en cada generación se hará de los mejores s individuos tomando en cuenta los individuos de la población actual y de la selección de la generación pasada. Esto se puede ver como elitismo, donde los s mejores individuos de las generaciones pasadas se mantienen en una memoria, sin necesidad de reemplazar ningún individuo de la población actual, pero también son tomados en cuenta al momento de realizar la selección en la generación actual. Este operador de selección es importante, pues al haber un componente estocástico, las mejores soluciones se pueden perder y nunca volverse a muestrear. Entonces mediante este operador de selección se asegura que las mejores soluciones no se pierdan en el proceso evolutivo.

3.2 Tau de Kendall

La versión de la medida de asociación, conocida como tau de Kendall [Kendall, 1938], basada en una muestra es definida en términos de concordancia:

Definición 3. Sea $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ una muestra aleatoria de n observaciones de un vector (X, Y) de variables aleatorias continuas. Hay $\binom{n}{2}$ pares distintos (x_i, y_i) y (x_j, y_j) de observaciones en la muestra, y cada par es concordante o discordante, sea c el número de pares concordantes y d el número de pares discordantes. Entonces, el tau de Kendall para la muestra está definido como:

$$\tau = \frac{c - d}{c + d} = \frac{c - d}{\binom{n}{2}}$$

Las cópulas arquimedeanas están relacionadas con el tau de Kendall. Genest y MacKay

[Genest and MacKay, 1986] demostraron que el tau de Kendall τ esta dado por

$$\tau = 4 \int_I \frac{\varphi(v)}{\varphi'(v)} dv + 1$$

Por lo anterior, un método para estimar el parámetro de las cópulas de dos variables es calcularlo a partir del tau de Kendall.

| Tipo de cópula | τ | θ |
|----------------|--|----------------------------|
| Clayton | $\frac{\theta}{(\theta + 2)}$ | $\frac{2\tau}{(1 - \tau)}$ |
| Gumbel | $1 - \frac{1}{\theta}$ | $\frac{1}{(1 - \tau)}$ |
| Frank | $1 - \frac{4}{\theta} (1 - D_1(\theta))$ | No hay formula cerrada |

En la tabla anterior $D_1(\theta)$ es la función Debye de primer orden [Abramowitz and Stegun, 1972].

No obstante la inexistencia de una formula cerrada para obtener el parámetro θ de la cópula de Frank, es posible obtener una aproximación mediante un algoritmo de búsqueda de raíces a partir del tau de Kendall, utilizando la función $1 - \frac{4}{\theta} (1 - D_1(\theta)) - \tau = 0$, donde ya conocemos el valor de τ y θ será la variable de la cual se encontrará la raíz.

3.3 Estimación de parámetros

El método de la máxima verosimilitud realiza la optimización de los parámetros de la cópula y de las distribuciones marginales al mismo tiempo, lo que puede ser computacionalmente muy intenso. Joe y Xu [Joe and Xu, 1996], propusieron que este conjunto de parámetros fuera estimado en dos pasos:

1. Como primer paso, se estiman los parámetros θ_1 de los marginales, realizando la estimación de las distribuciones marginales de una variable:

$$\hat{\theta}_1 = \arg \max_{\theta_1} \sum_{t=1}^T \sum_{j=1}^n \ln f_j(x_{jt}; \theta_1)$$

Para nuestros fines, en este paso se hará lo siguiente:

- a) Para cada variable X_i , estimar los parámetros de los marginales beta (α_i, β_i) para $i = 1, \dots, n$ por el método de momentos, el cual utiliza los valores de la media y varianza de la muestra.
- b) Determinar $U_i = F_i(X_i; \alpha_i, \beta_i)$ para $i = 1, \dots, n$. Donde U_i es la variable X_i expresada mediante la distribución marginal beta con parámetros (α_i, β_i) .

2. Como segundo paso, dado θ_1 , se realiza la estimación de los parámetros θ_2 de la cópula:

$$\hat{\theta}_2 = \arg \max_{\theta_2} \sum_{t=1}^T \ln c \left(F_1(x_{1t}), \dots, F_n(x_{nt}); \theta_2, \hat{\theta}_1 \right)$$

Este método de parámetros hace que se simplifique el proceso de estimación, al realizarla para cada distribución marginal y de los parámetros de la cópula por separado. Esto aunque aumenta el número de estimaciones a realizar, cada una significa un problema computacionalmente mas sencillo y menos intensivo.

3.3.1 Estimación incremental de parámetros

Debido a la especificidad que presentan los parámetros en generaciones tempranas del algoritmo, lo que significa una convergencia prematura de este al quedar atrapado en un posible óptimo local, y para evitar la necesidad de muestras de tamaño mayor en el proceso de estimación que eviten lo antes mencionado, lo que significaría un mayor costo computacional al incrementar el tiempo en el computo en los procesos que dependen del tamaño de la muestra como el calculo del Kendall de tau y la verosimilitud de una cópula, se utiliza un método de sumas ponderadas que incrementalmente estima los parámetros tomando en cuenta los valores de los parámetros obtenidos en generaciones previas. El método mencionado fue propuesto en [Sebag and Ducoulombier, 1998] y consiste en las

siguientes formulas:

$$\mu_{t+1} = (1 - \alpha)\mu_t + \alpha\bar{x}$$

$$\sigma_{t+1}^2 = (1 - \alpha)\sigma_t^2 + \alpha\left(\frac{1}{N-1}\sum_{i=1}^N(x_i - \bar{x})^2\right)$$

donde α es un parámetro en el intervalo $[0, 1]$.

Las formulas anteriores ayudan por un lado a mover la media a un paso menos acelerado hacia los mejores individuos, y por el otro a que la varianza no se haga pequeña a una velocidad muy rápida.

El parámetro α es de gran importancia, ya que su valor determina el paso o la rapidez con que los parámetros se mueven o se especifican. Para el calculo automático de este parámetro se probaron tres formulas mediante una metodología no rigurosa:

1. $\alpha_g = \frac{1}{\log(Gen - g + 3)}, \quad g = 1, 2, \dots, Gen$
2. $\alpha_g = \frac{g}{Gen}, \quad g = 1, 2, \dots, Gen$
3. $\alpha_g = \frac{g}{Gen} \frac{1}{\log(Gen - g + 3)}, \quad g = 1, 2, \dots, Gen$

Las pruebas llevadas a cabo mostraron que la primer formula mostraba mejores resultados comparadas con las otras 2 formulas.

3.4 Modelo gráfico probabilístico

La representación de un modelo gráfico probabilístico consiste de dos componentes:

- Estructura. Es un grafo acíclico dirigido que representa un conjunto de (in)dependencias condicionales en las variables de X .
- Conjunto de distribuciones de probabilidad generalizadas locales. Esto es, las probabilidades de cada variable dadas las probabilidades de las variables con las que esta directamente relacionada.

3.4.1 Red bayesiana de cópulas

Una red bayesiana, como ya vimos, es un modelo gráfico probabilístico que sirve para representar variables discretas. Una red bayesiana de cópulas es igualmente un modelo gráfico probabilístico, con una estructura de grafo acíclico dirigido, pero con la particularidad que las funciones de probabilidad locales son cópulas, lo que significa que puede representar variables continuas.

Este modelo gráfico probabilístico recientemente propuesto en [Elidan, 2010], muestra gran promesa al hacer uso de la estructura de una red bayesiana, misma que ha demostrado obtener resultados de gran calidad en EDAs para optimización combinatoria, al igual que su contraparte para optimización numérica, mediante las redes gaussianas, las cuales limitan la distribución que pueden describir las variables y las dependencias entre éstas a la distribución gaussiana. Sin embargo, este tipo de estructura no había sido utilizada en EDAs para optimización numérica que utilizan cópulas, las cuales abren la posibilidad de describir las variables con diferentes distribuciones marginales, así como de las dependencias entre variables que pueden ser descritas, al haber una variedad de cópulas.

La pieza principal de las redes bayesianas es una densidad condicional local. La parametrización de tal densidad usando cópulas es:

Lema 1. [Elidan, 2010] Sea $f(x | \mathbf{y})$, con $\mathbf{y} = \{y_1, \dots, y_k\}$, una función de densidad condicional y sea $f(x)$ la densidad marginal de X . Entonces existe una función de densidad de cópula tal que

$$f(x | \mathbf{y}) = R_c(F(x), F(y_1), \dots, F(y_k))$$

donde R_c es la razón

$$R_c(F(x), F(y_1), \dots, F(y_k)) \equiv \frac{c(F(x), F(y_1), \dots, F(y_k))}{\int c(F(x), F(y_1), \dots, F(y_k)) f(x) dx} = \frac{c(F(x), F(y_1), \dots, F(y_k))}{\frac{\partial^k C(1, F(y_1), \dots, F(y_k))}{\partial F(y_1) \dots \partial F(y_k)}}$$

y donde R_c se define como 1 cuando $\mathbf{y} = \emptyset$. Lo opuesto también se cumple, para cualquier función de densidad de cópula c , $R_c(F(x), F(y_1), \dots, F(y_k))$ define una función de densidad condicional válida.

Sea \mathcal{G} un grafo dirigido acíclico cuyos nodos corresponden a las variables aleatorias \mathbf{X} , y sea $Pa_i = \{Pa_{i1}, \dots, Pa_{ik}\}$ los padres de X_i en \mathcal{G} . \mathcal{G} codifica las declaraciones de independencia $I(\mathcal{G}) = \{(X_i \perp NoDescendientes_i \mid Pa_i)\}$, donde $NoDescendientes_i$ son nodos que no son descendientes de X_i en \mathcal{G} . Se dice que $f_{\mathbf{X}}(\mathbf{x})$ se descompone de acuerdo con \mathcal{G} si puede ser escrita como un producto de densidades condicionales $f_{\mathbf{X}}(\mathbf{x}) = \prod_i f(X_i \mid Pa_i)$. Si f se descompone de acuerdo con \mathcal{G} entonces $I(\mathcal{G})$ se cumple en $f_{\mathbf{X}}(\mathbf{x})$. La inversa es verdad: si $I(\mathcal{G})$ se cumple en $f_{\mathbf{X}}(\mathbf{x})$ entonces la densidad se descompone de acuerdo con \mathcal{G} . Estos resultados son la base del modelo de red bayesiana donde una densidad conjunta es construida vía la descomposición de densidades locales. Lo anterior, a la vez, provee las bases para la construcción del modelo de red bayesiana de cópulas.

Teorema 3. [Elidan, 2010] *Descomposición.* Sea \mathcal{G} un grafo dirigido acíclico sobre \mathbf{X} , y esté $f_{\mathbf{X}}(\mathbf{x})$ parametrizada vía una densidad de cópula conjunta $f_{\mathbf{X}}(\mathbf{x}) = c(F(x_1), \dots, F(x_n)) \prod_i f(x_i)$, con $f_{\mathbf{X}}(\mathbf{x})$ estrictamente positiva para todos los valores de \mathbf{X} . Si $f_{\mathbf{X}}(\mathbf{x})$ se descompone de acuerdo con \mathcal{G} entonces la densidad de la cópula $c(F(x_1), \dots, F(x_n))$ también se descompone de acuerdo con \mathcal{G}

$$c(F(x_1), \dots, F(x_n)) = \prod_i R_{c_i}(F(x_i), \{F(pa_{ik})\})$$

donde c_i es una cópula local que depende solo del valor X_i y sus padres en \mathcal{G} .

Teorema 4. [Elidan, 2010] *Composición.* Sea \mathcal{G} un grafo dirigido acíclico sobre \mathbf{X} . Adicionalmente, sea $\{c_i(F(x_i), F(pa_{i1}), \dots, F(pa_{ik}))\}$ un conjunto de densidad de cópulas estrictamente positivas asociadas con los nodos de \mathcal{G} que tienen al menos un padre. Si $I(\mathcal{G})$ se cumple entonces la función

$$g(F(x_1), \dots, F(x_n)) = \prod_i R_{c_i}(F(x_i), \{F(pa_{ik})\})$$

es una densidad de cópula $c(F(x_1), \dots, F(x_n))$ válida sobre \mathbf{X} .

Es importante notar que las cópulas locales no necesitan concordar en los marginales univariados de las variables traslapadas, lo que nos da la libertad de mezclar y combinar cópulas de diferentes tipos.

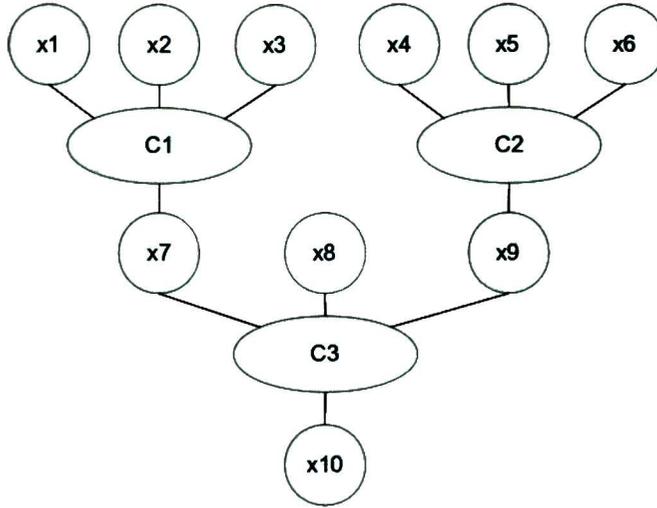


Figura 3.1: Red bayesiana de cópulas

Todo lo anterior permite la construcción de una densidad conjunta, dados marginales univariados, mediante la adecuada composición de términos locales sin preocuparse de coherencia global:

Definición 4. Una red bayesiana de cópulas es una tripleta $\mathcal{C} = (\mathcal{G}, \Theta_C, \Theta_f)$ que codifica la densidad conjunta $f_{\mathbf{X}}(\mathbf{x})$. Θ_C es un conjunto de funciones de densidad locales $c_i(F(x_i), \{F(pa_{ik})\})$ que están asociadas con los nodos de \mathcal{G} que tienen al menos un padre. Θ_f es el conjunto de parámetros que representan las densidades marginales $f(x_i)$. $f_{\mathbf{X}}(\mathbf{x})$ esta parametrizada como

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_i R_{c_i}(F(x_i), \{F(pa_{ik})\}) f(x_i)$$

La estructura de una red bayesiana de cópulas se puede apreciar en la figura 3.1.

Para mayor información acerca de las redes bayesianas de cópulas y la demostración de los teoremas 3 y 4, y del lema 1, referirse a [Elidan, 2010].

3.4.2 Funciones de densidad

Se utiliza la forma $c = \varphi^{-(n)}(C(u_1, \dots, u_n)) \prod_{i=1}^n \varphi'(u_i)$ para representar la densidad de las cópulas.

A continuación se describen las tres cópulas de la familia arquimedea utilizadas en este trabajo. Estas cópulas fueron seleccionadas debido a que cada una representa un tipo diferente de dependencia, además de que son tres de las cópulas más representativas de la familia arquimedea y las únicas de las cuales se encontró una forma multivariable en la literatura revisada. Otra ventaja muy útil para este trabajo es, que debido a la utilización de las densidades de las cópulas para calcular las densidades locales y la obtención de éstas densidades mediante la n derivada de la inversa del generador de cada cópula, existen formulas recursivas para la obtención de estas derivadas, lo que permite extender la cantidad de variables que puede tener una cópula según se desee.

3.4.2.1. Cópula de Frank

La n -ésima derivada de la función inversa del generador de la cópula de Frank tiene la forma:

$$\varphi^{-1(n)}(t) = -\frac{1}{\theta} \Psi_{n-1} \left(\frac{1}{1 + e^{-t}(e^{-\theta} - 1)} \right)$$

donde $\Psi_{n-1}(y)$ es una función de y que puede ser determinada recursivamente, empezando con $\Psi_0(y) = y - 1$, como sigue:

$$\Psi_n(y) = y(1-y)\Psi'_{n-1}(y)$$

$$\Psi_1(y) = -y^2 + y$$

$$\Psi_2(y) = 2y^3 - 3y^2 + y$$

$$\Psi_3(y) = -6y^4 + 12y^3 - 7y^2 + y$$

$$\Psi_4(y) = 24y^5 - 60y^4 + 50y^3 - 15y^2 + y$$

$$\Psi_5(y) = -120y^6 + 360y^5 - 390y^4 + 180y^3 - 31y^2 + y$$

Por lo tanto, la densidad de la cópula de Frank puede ser calculada mediante la función:

$$c = -\frac{1}{\theta} \Psi_{n-1} \left(\frac{1}{1 + e^{-x}(e^{-\theta} - 1)} \right) \prod_{i=1}^n \varphi'(u_i)$$

donde $x = \sum_{i=1}^n \varphi(u_i)$ y $\varphi'(u) = \frac{\theta}{1 - e^{\theta u}}$

3.4.2.2. Cópula de Gumbel

En este caso se utiliza una parametrización diferente $\alpha = \theta^{-1}$. La n -ésima derivada de la función inversa del generador de la cópula de Gumbel tiene la forma:

$$\varphi^{-1(n)}(t) = (-1)^n \alpha \exp(-t^\alpha) t^{-n+\alpha} \Psi_{n-1}(t^\alpha)$$

donde $\Psi_{n-1}(y)$ es una función de y que puede ser determinada recursivamente, empezando con $\Psi_0(y) = 1$, como sigue:

$$\Psi_n(y) = [\alpha(y-1) + n] \Psi_{n-1}(y) - \alpha y \Psi'_{n-1}(y)$$

$$\Psi_1(y) = \alpha(y-1) + 1$$

$$\Psi_2(y) = \alpha^2(y^2 - 3y + 1) + 3\alpha(y-1) + 2$$

$$\Psi_3(y) = \alpha^3(y^3 - 6y^2 + 7y - 1) + 6\alpha^2(y^2 - 3y + 1) + 11\alpha(y-1) + 6$$

$$\Psi_4(y) = \alpha^4(y^4 - 10y^3 + 25y^2 - 15y + 1) + 10\alpha^3(y^3 - 6y^2 + 7y - 1) + 35\alpha^2(y^2 - 3y + 1) + 50\alpha(y-1) + 24$$

$$\Psi_5(y) = \alpha^5(y^5 - 15y^4 + 65y^3 - 90y^2 + 31y - 1) + 15\alpha^4(y^4 - 10y^3 + 25y^2 - 15y + 1) + 85\alpha^3(y^3 - 6y^2 + 7y - 1) + 225\alpha^2(y^2 - 3y + 1) + 274\alpha(y-1) + 120$$

Por lo tanto, la densidad de la cópula de Gumbel puede ser calculada mediante la función:

$$c = (-1)^n \alpha \exp(-x^\alpha) x^{-n+\alpha} \Psi_{n-1}(x^\alpha) \prod_{i=1}^n \varphi'(u_i)$$

$$\text{donde } x = \sum_{i=1}^n (-\ln(u_i))^{\frac{1}{\alpha}} \text{ y } \varphi'(u) = \frac{(-\ln u)^{1/\alpha}}{\alpha u \ln u}.$$

3.4.2.3. Cópula de Clayton

La n -ésima derivada de la función inversa del generador de la cópula de Gumbel tiene la forma:

$$\varphi^{-1(n)}(t) = (-1)^n (1 + \theta t)^{-(1+n\theta)/\theta} \prod_{i=0}^{n-1} (1 + i\theta)$$

Por lo tanto, la densidad de la cópula de Clayton puede ser calculada mediante la función:

$$c = (-1)^n (1 + \theta x)^{-(1+n\theta)/\theta} \prod_{i=0}^{n-1} (1 + i\theta) \prod_{i=1}^n \varphi'(u_i)$$

donde $x = \sum_{i=1}^n \frac{(u_i^{-\theta} - 1)}{\theta}$ y $\varphi'(u) = -u^{-\theta-1}$.

3.4.3 Aprendizaje de la estructura del grafo

En este trabajo se utiliza un método de *score+search*, dado que el modelo probabilístico esta claramente expresado mediante su función de densidad, la cual es directamente utilizada por este tipo de métodos en las medidas de verosimilitud del modelo. Se utiliza un algoritmo de búsqueda voraz en conjunto con métricas para estimar la aptitud de la estructura.

La métrica utilizada en esta propuesta es la BIC [Schwarz, 1978] la cual hace un balance entre la verosimilitud del modelo y la complejidad de este. Es la cualidad de la métrica BIC que castiga la complejidad del modelo en cuestión la que la hace tan útil y la razón por la que fue elegida, pues indudablemente la adición de variables al modelo significa un incremento en su verosimilitud, sin embargo esto hace que el modelo se haga mas complejo de lo que necesita ser, tomando en cuenta dependencias que son muy débiles y podrían ser ignoradas o pasadas por alto, y por lo tanto aumentando el costo computacional que conlleva la estimación de las densidades locales y el muestreo a partir de este modelo mas complejo de lo necesario. La métrica BIC tiene la siguiente forma:

$$\text{score}(\mathcal{G} : \mathcal{D}) = l(\mathcal{D} : \hat{\theta}_{\mathcal{G}}) - \frac{1}{2} \ln(M) |\Theta_{\mathcal{G}}|$$

donde, \mathcal{D} son los datos observados, $\hat{\theta}$ son los parámetros estimados mediante el método de la máxima verosimilitud, $l(\mathcal{D} : \hat{\theta}_{\mathcal{G}})$ es la verosimilitud logarítmica de los parámetros y la estructura dados los datos observados, M es el numero de datos observados y $|\Theta_{\mathcal{G}}|$ es el número de parámetros asociados con la estructura del grafo \mathcal{G} , el cual es igual a 1 por cada cópula arquimedean de un parámetro y los parámetros de los marginales no se cuentan pues están fijos.

La maquinaria que se utiliza es descrita en [Bosman and Thierens, 2000a], en la sección 3.2. La maquinaria consiste, en un inicio, de un grafo vacío. Posteriormente se prueba arista por arista hasta

encontrar aquella que agregue más información a la estructura de acuerdo con la métrica que se utilice (en nuestro caso la BIC). Una vez agregada la arista se eliminan las aristas que puedan crear ciclos dentro del grafo y se eliminan de la búsqueda para poder agregarlas en un futuro.

Se propone acelerar esta maquinaria de aprendizaje mediante una heurística que se basa en las siguientes ideas:

- El uso del tau de Kendall como medida para agregar una nueva arista a la estructura del modelo probabilístico, en vez de calcular la verosimilitud por cada arista que se prueba hasta obtener la que agrega más información al modelo como se hace en la maquinaria original. Se utiliza una matriz con los coeficientes del tau de Kendall por cada par de variables para identificar correlaciones entre variables y así obtener las nuevas posibles aristas en la estructura que signifiquen la creación de una cópula nueva en el modelo probabilístico. Para esto, se buscan coeficientes positivos, puesto que la estructura sólo permite dependencias positivas. Además, también se busca la posible adición de aristas a la estructura que signifiquen la adición de una variable a una cópula ya existente. Este tipo de adición se mide mediante el promedio de los tau de Kendall para cada par de variables existentes en la cópula y de la nueva variable que se desea agregar. Igualmente se busca que el promedio del tau de Kendall sea positivo. Una vez obtenida una nueva posible arista en el grafo, se utiliza la métrica BIC [Schwarz, 1978] para decidir si la adición de la nueva arista realmente incrementa la aptitud de la estructura del modelo probabilístico.
- La estimación del parámetro de la cópula, durante el aprendizaje del grafo, a partir del tau de Kendall. Puesto que la asociación del tau de Kendall para la estimación de los parámetros de las cópulas arquimedeanas no se extiende al caso multivariable, se hace uso de la propiedad asociativa de las cópulas arquimedeanas para estimar el parámetro de la cópula multivariable a partir del valor promedio de los valores del tau de Kendall de cada par de variable presentes en la cópula, lo que significaría una alternativa menos costosa computacionalmente comparada con la estimación del parámetro de la cópula mediante el método de la máxima verosimilitud,

Algorithm 1 Búsqueda de grafo condicional

```

BUSQUEDA_DE_GRAFO_CONDICIONAL ()
 $a \leftarrow$  matriz booleana nueva de 2 dimensiones con tamaño  $l \times l$ 
 $v^p, v^s \leftarrow$  2 arreglos nuevos de (vectores de enteros) con tamaño  $l$ 
de  $i \leftarrow 0$  a  $l - 1$  hacer
  de  $j \leftarrow 0$  a  $l - 1$  hacer
     $a[i, j] \leftarrow$  verdadero
     $a[i, i] \leftarrow$  falso
 $\gamma \leftarrow l^2 - l$ 
mientras que  $\gamma > 0$  hacer
   $(v_0, v_1) \leftarrow$  ENCONTRAR_ARISTA_PROPUESTA ( $a, v^p, v^s$ )
  si  $v_0 < 0$  entonces
    terminar ciclo
   $\gamma \leftarrow \gamma -$  AGREGAR_ARCO ( $\kappa, v_0, v_1, a, v^p, v^s$ )
regresar(ORDEN_TOPOLOGICO ( $v^p$ ))

```

- La selección de la cópula, de las tres utilizadas en este trabajo, que refleja mejor el tipo de dependencia que existe entre las variables al momento de la creación de una cópula nueva en el modelo probabilístico. Esto se hace mediante la selección de la cópula con mayor verosimilitud, la cual se mantiene fija a través del proceso.

El proceso de aprendizaje con la heurística propuesta se puede ver en los algoritmos 1 y 2. El algoritmo 1 trata de encontrar aristas para agregar al grafo; para llevar a cabo esto hace uso de una matriz donde registra las aristas que pueden ser añadidas y las que no, además de dos listas de adyacencia del grafo, una de los padres y otra de los hijos de cada variable, si es que existen. En el algoritmo 2, se realiza la búsqueda de las aristas por añadir al grafo mediante la aceleración propuesta que hace uso del tau de Kendall; aquí se hace uso de la matriz de aristas posibles para saber si se probará una arista, y de la lista de adyacencia de padres e hijos de cada variable para saber si la arista que se está probando creará una nueva cópula o añadirá una variable a una cópula existente. A continuación se describe el funcionamiento de los métodos no detallados en los algoritmos:

- ENCONTRAR_MEJOR_COPULA: Este método compara la densidad obtenida, utilizando la muestra seleccionada y el parámetro estimado a partir del tau de Kendall, de las tres cópulas utilizadas y selecciona como la mejor cópula la que tenga la mayor densidad.

Algorithm 2 Encontrar arista por métrica con propuesta

 ENCONTRAR_ARISTA_PROPUESTA (a, v^p, v^s)

 $\delta_{max} \leftarrow 0$ $i_{max} \leftarrow -1$ $j_{max} \leftarrow -1$ **de** $i \leftarrow 0$ **a** $l - 1$ **hacer** **de** $j \leftarrow 0$ **a** $l - 1$ **hacer** **si** $a[i, j]$ **entonces** **si** $\tau[i, j] > \delta_{max}$ **entonces** **si** tamaño de $v^p[j] = 0$ **entonces** $\delta_{max} \leftarrow \tau[i, j]$ $i_{max} \leftarrow i$ $j_{max} \leftarrow j$ **por cada** cópula **hacer** $\tau_{promedio} \leftarrow 0$ **de** $i \leftarrow 0$ **a** tamaño de $v^p[j_{max}]$ **hacer** $\tau_{promedio} = \tau_{promedio} + \tau(v^p[i])$ **de** $i \leftarrow 0$ **a** $l - 1$ **hacer** **si** $a[i, \text{variable hijo en la cópula}]$ **entonces** **si** $((\tau_{promedio} + \tau(i, \text{variable hijo en la cópula})) / (\text{tamaño de } v^p[\text{variable hijo en la cópula}] + 1)) > \delta_{max}$ **entonces** $\delta_{max} \leftarrow ((\tau_{promedio} + \tau(i, \text{variable hijo en la cópula})) / (\text{tamaño de } v^p[\text{variable hijo en la cópula}] + 1))$ $i_{max} \leftarrow i$ $j_{max} \leftarrow \text{variable hijo en la cópula}$ **si** $\delta_{max} = 0$ **entonces** $i_{max} \leftarrow -1$ $j_{max} \leftarrow -1$ **en caso contrario** **si** tamaño de $v^p[j_{max}] = 0$ **entonces** $C \leftarrow \text{ENCONTRAR_MEJOR_COPULA}(i_{max}, j_{max})$ $\text{COPULAS}_{\text{COPULAS}} \leftarrow C$ **en caso contrario** **si no** **AUMENTA_METRICA** (i_{max}, j_{max}) **hacer** $a(i_{max}, j_{max}) \leftarrow \text{falso}$ $(i_{max}, j_{max}) \leftarrow \text{ENCONTRAR_ARISTA_PROPUESTA}(a, v^p, v^s)$ **regresar** ((i_{max}, j_{max}))

- **AUMENTA_METRICA:** Este método verifica que al agregar una arista al grafo se aumente el valor de la métrica utilizada.
- **AGREGAR_ARCO:** Este método agrega una arista al grafo y actualiza las estructuras y métricas utilizadas para representar el grafo.
- **ORDEN_TOPOLOGICO:** Este método devuelve el orden de muestreo de las variables.

3.4.4 Estimación de parámetros finales de las cópulas

Una vez terminado el aprendizaje de la estructura del grafo, se calculan los parámetros de las cópulas que conforman el grafo, mediante el método de la máxima verosimilitud. Se utiliza el método quasi Newton para obtener el parámetro que maximiza la densidad de la cópula a partir de la muestra que se tiene. Se eligió este método por ser uno de los que presentan convergencia más rápida. En la práctica comprobamos que no se requiere de métodos más poderosos que consuman mayor tiempo de cómputo.

3.5 Muestreo

Es utilizado un método general para simular muestras de una cópula, el cual utiliza un enfoque condicional. El procedimiento general para un escenario de multivariable es como sigue:

- Simular una variable aleatoria u_1 de $U(0, 1)$.
- Simular una variable aleatoria u_2 a partir de $C_2(\cdot | u_1)$
- Continuar el proceso de simulación agregando variables condicionales hasta simular una variable aleatoria u_n a partir de $C_n(\cdot | u_1, \dots, u_{n-1})$

Para simular un valor u_k a partir de $C_k(\cdot | u_1, \dots, u_{k-1})$ se tiene que generar una variable aleatoria v de $U(0, 1)$, de la cual $u_k = C_k^{-1}(v | u_1, \dots, u_{k-1})$ puede ser obtenida a partir de la ecuación $v = C_k(u_k | u_1, \dots, u_{k-1})$ mediante un algoritmo de búsqueda raíces.

Sea $C(u_1, u_2, \dots, u_n) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_n))$ una cópula arquimedea de n variables con generador $\varphi(\cdot)$ y con $k = 2, \dots, n$, el método de muestreo puede reescribirse de la siguiente manera:

$$C_k(u_k | u_1, \dots, u_{k-1}) = \frac{\varphi^{-1(k-1)}(c_k)}{\varphi^{-1(k-1)}(c_{k-1})}$$

donde $c_k = \sum_{j=1}^k \varphi(u_j)$ y $\varphi^{-1(k)}(\cdot)$ es la k -ésima derivada de la inversa del generador.

Se utilizan las siguientes reglas para decidir a partir de qué cópula se muestreará una variable, puesto que una variable puede pertenecer a muchas cópulas.

- Si la variable es hijo en una cópula, se utilizará esta cópula para muestrear la variable.
- Si la variable no es hijo en ninguna cópula, se verificará a cuántas cópulas pertenece y se muestreará a partir de la cópula en la cual ya se hayan muestreado la mayor cantidad de variables que componen a la cópula, y resten por muestrear menos variables. En caso de que dos cópulas queden empatadas, se utilizará la que tenga mayor parámetro dado que el rango del parámetro de las cópulas utilizadas es de $(0, \infty)$, exceptuando el parámetro la cópula de Gumbel que tiene un rango de $(1, \infty)$ al cual se le restará 1 para ponerlo en el mismo rango para poder compararlo con las otras cópulas.
- Si la variable no pertenece a ninguna cópula, se muestreará aleatoriamente.

3.6 Resumen

En este capítulo se presentaron los bloques con los que se construye el algoritmo propuesto en este trabajo. Primero vimos el tau de Kendall como una medida de correlación que no supone distribución alguna y su valor está relacionado con la estimación de los parámetros de cópulas bivariadas. Después vimos un método de estimación de modelos multivariados donde los parámetros de los marginales univariados y de la función de densidad conjunta se estiman por separado, esto para hacer menos intensivo el proceso de estimación. Se presentó también el modelo probabilístico

utilizado por el algoritmo propuesto, el cual tiene la estructura de una red bayesiana con la diferencia que las funciones de densidad locales son cópulas, lo que le permite describir relaciones entre variables continuas. Se utilizan las cópulas de Frank, de Gumbel y de Clayton, las cuales pertenecen a la familia arquimedea, como funciones locales en la red bayesiana de cópulas. Estas cópulas permiten el uso del tau de Kendall para proponer una heurística que acelera el aprendizaje de la estructura de la red, que se hace mediante un algoritmo voraz y la métrica BIC como medida de la estructura, utilizando el valor del tau de Kendall como medida para agregar una arista y para estimar a partir de él el parámetro de las cópulas multivariadas. Por último se vio un método de muestreo condicional y un conjunto de reglas para seleccionar la cópula de la cual se muestreará una variable, en caso de que esta pertenezca a varias cópulas.

4

Resultados y evaluación

A continuación se presentan los detalles de la implementación del algoritmo propuesto, así como de los experimentos llevados a cabo y los resultados obtenidos de éstos.

4.1 Implementación

Se realizó la implementación en lenguaje C++.

Se implementó una clase base con los atributos comunes entre las 3 cópulas arquimedeanas que se utilizan en la propuesta, que son las de Frank, de Gumbel y de Clayton. Los atributos comunes son:

- Las variables de las que se componen las cópulas (padres e hijo)
- El valor del parámetro
- El cociente actual de la cópula R_c

Además de esto, se tienen atributos que corresponden a los valores que se obtienen al verificar si una variable se agregará a una cópula, esto para que en caso de agregarse la variable a la cópula no se

vuelvan a realizar los mismo cálculos de estimación del parámetro de la cópula y del cociente R_{c_i} de la cópula. Estos atributos auxiliares son:

- La variable que posiblemente se agregará.
- El nuevo cociente calculado.
- El nuevo valor del parámetro de la cópula.
- El tau de Kendall a partir del cual se estimó el parámetro de la cópula.

A partir de esta clase base, se crearon 3 clases para representar cada una de la cópulas, las cuales heredan los atributos declarados en la clase base, además de implementar los métodos específicos para los cálculos de:

- La verosimilitud logarítmica
- Calculo del cociente R_{c_i}
- Estimación del parámetro de la cópula por el método de la máxima verosimilitud. Se implementó el método quasi Newton con un mecanismo de reinicio del punto de búsqueda, en caso de que se encuentre un punto de inflexión que no sea máximo o el punto se salga de la región factible, dependiendo de la cópula para la que se este utilizando.
- El cálculo de la densidad de la cópula.
- La estimación del parámetro de la cópula a partir del tau de Kendall. Se implementó el método quasi Newton para la estimación del parámetro de la cópula de Frank a partir del tau de Kendall, ya que es la única que no cuenta con una función directa para esto.

Además, se implementó la maquinaria estándar de aprendizaje del grafo de la red bayesiana de cópulas, a la cual se agregó el enfoque propuesto de aprendizaje a partir del tau de Kendall y una estrategia para poder calcular la verosimilitud de la estructura, la cual consiste en mantener los grupos de cópulas y variables que pertenecen a un grafo conexo, dentro de la estructura del modelo probabilístico.

La impresión de los resultados generacionales y globales se realiza en formato XML, el cual muestra los parámetros utilizados para la ejecución del algoritmo, y la mejor solución y las estadísticas de uso de las cópulas obtenidas en cada generación y globalmente.

Se hizo uso de la librería boost para C++, en específico del paquete de distribuciones estadísticas y funciones especiales, el cual cuenta con una variedad de distribuciones estadísticas univariadas y funciones que operan sobre estas distribuciones.

4.2 Experimentos

Se utilizó el algoritmo propuesto para optimizar las funciones propuestas en [Tang et al., 2010]. En este *benchmark* se propone el uso de seis funciones como base:

- Función esfera
- Función elíptica rotada
- Función de Rastrigin rotada
- Función de Ackley rotada
- Problema 1.2 de Schwefel
- Función de Rosenbrock

Estas funciones base son modificadas de tal modo que se presenten cuatro tipos de problemas:

- Funciones separables
- Funciones con un número pequeño de variables dependientes entre sí y las variables restantes son independientes
- Funciones con múltiples subconjuntos de variables no separables entre sí, pero independientes entre subconjuntos

- Funciones completamente no separables

De tal modo que el número total de funciones propuestas es de 20. Todos los problemas suponen la minimización de la función, donde el óptimo global de todas las funciones es igual a 0.

4.3 Experimentos variando el parámetro de relajación

Los experimentos se llevaron a cabo sobre las funciones 1-3 y 19-20 de las 20 funciones de las que se compone el *benchmark* utilizado.

Para la resolución de cada problema se realizaron un total de 31 corridas, debido al componente estocástico del algoritmo.

En cada una de las 31 corridas que se llevaron a cabo, para la optimización de cada una de las funciones utilizadas, se utilizaron los mismos parámetros para tener un conjunto de corridas comparables entre si, de las cuales se pudieran obtener estadísticas relevantes.

Los parámetros utilizados en la experimentación son:

- Población: 100 individuos
- Número máximo de generaciones: 1000 (máximo 100000 evaluaciones)
- Número máximo de padres: 5 (lo que supone cópulas con un total de 6 variables)
- Porcentaje de selección: 30
- Dimensionalidad de los problemas: 10 variables de decisión

Además, para cada función se llevaron a cabo experimentos con un conjunto de valores para el parámetro de relajación de las distribuciones marginales, con los cuales se realizaron al igual 31 corridas para cada valor del parámetro de relajación. El conjunto de valores con los cuales se llevaron a cabo los experimentos es: $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 1\}$.

Estos experimentos se llevaron a cabo con el fin de entender el comportamiento del algoritmo con y sin la utilización del parámetro de relajación. Se seleccionó el valor 0.01 como el valor del

| Parámetro de relajación | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-------------------------|--------------|---------------|--------------|--------------|--------------|--------------|
| Media | 0.0979565539 | 31.2896451613 | 0.2902682581 | 0.0255343 | 0.0156041323 | 0.01229602 |
| Desviación Estándar | 0.4219273472 | 8.6869257698 | 0.0780850698 | 0.0065757558 | 0.0043624762 | 0.0031375432 |
| Varianza | 0.1780226863 | 75.4626793299 | 0.0060972781 | 4.32405E-005 | 1.90312E-005 | 9.84418E-006 |
| Máximo | 2.36309 | 56.996 | 0.418131 | 0.0388623 | 0.0252196 | 0.019836 |
| Mínimo | 0.00257028 | 11.3821 | 0.1043 | 0.0137054 | 0.00537987 | 0.00719508 |

| Parámetro de relajación | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
|-------------------------|--------------|--------------|---------------|--------------|--------------|
| Media | 0.0091615945 | 0.0074050326 | 0.00681911 | 0.0062054177 | 0.0060614539 |
| Desviación Estándar | 0.0028283551 | 0.0023520686 | 0.0021530858 | 0.0016893274 | 0.0021301646 |
| Varianza | 7.99959E-006 | 5.53223E-006 | 4.635785E-006 | 2.85383E-006 | 4.53760E-006 |
| Máximo | 0.0147719 | 0.0149488 | 0.0106269 | 0.00898112 | 0.00976279 |
| Mínimo | 0.00375094 | 0.00351443 | 0.00351515 | 0.00290582 | 0.00264332 |

Tabla 4.1: Función 1

parámetro en las primeras pruebas que se realizaron, pues es el valor utilizado en la propuesta original[Sebag and Ducoulombier, 1998]. Sin embargo, al ver que los resultados no mejoraban notablemente en comparación con los resultados cuando no se utiliza el parámetro de relajación, se decidió empezar a variar el parámetro utilizando valores con un aumento pequeño al valor inicial. Al empezar a variar el parámetro se pudo notar que se mejoraban los resultados obtenidos, sin embargo se decidió dejar de experimentar hasta cierto punto donde los resultados eran de buena calidad y obtener una idea del valor del parámetro con el que el algoritmo arrojaba resultados aceptables, y a partir de eso poder proponer una heurística para el calculo automático del parámetro.

En la tabla 4.1 se reportan las estadísticas para los valores de la función alcanzados por el algoritmo en la función 1 del *benchmark* utilizado. Se puede observar que el mínimo obtenido cuando el parámetro de relajación de las distribuciones marginales es igual a uno, es decir cuando no existe relajación de los parámetros de las distribuciones marginales, es el menor de los mínimos obtenidos por todos los valores del parámetro de relajación con los que se experimento; sin embargo, la media de las 31 corridas con este mismo valor no es la mínima, y además la desviación estándar y varianza tienen los segundos valores mas altos, únicamente superados cuando el valor del parámetro de relajación es de 0.01. Además de esto, también se puede observar que en este caso, al acercarse el valor del parámetro de relajación de las distribuciones marginales a 0.1, el valor de la media, de la varianza y de la desviación estándar se hacen cada vez mas pequeños.

En la figura 4.1 se muestra un diagrama de caja con los resultados de las 31 corridas para cada valor del parámetro de relajación, optimizando la función 1. Este diagrama de caja tiene una

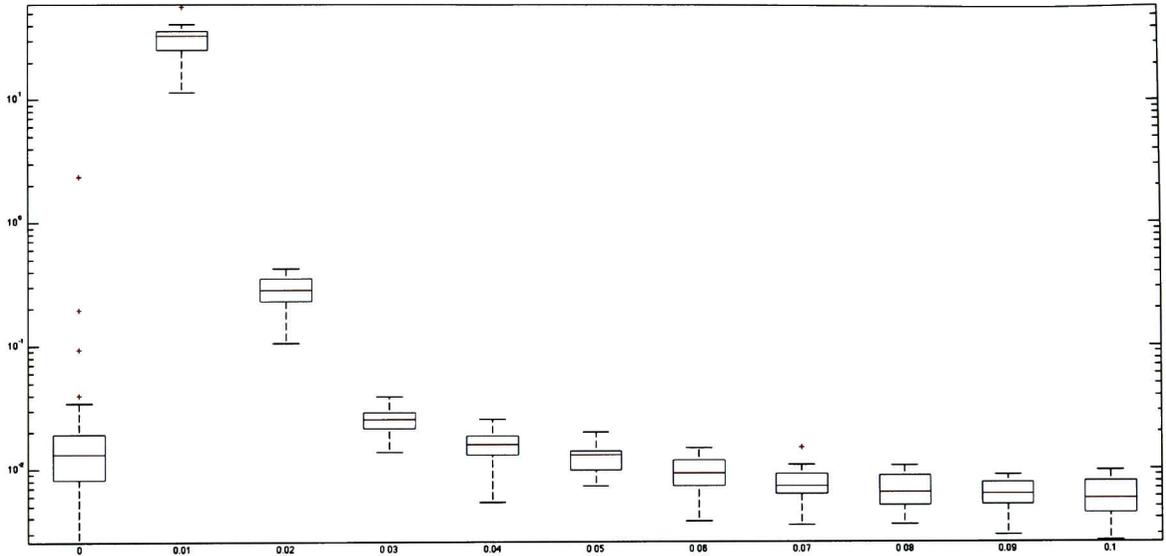


Figura 4.1: Función 1

escalación logarítmica en el eje de las Y, para poder mostrar los resultados de una manera que se aprecien mejor. En este diagrama de caja se puede apreciar que mediante la utilización de la estrategia de relajación de los parámetros de las distribuciones marginales, los experimentos son más estables, es decir, los experimentos nos darán como resultado soluciones dentro de un rango típico y raramente se obtendrán soluciones con valores muy alejados de este rango típico.

En la tabla 4.2 se reportan las estadísticas para los valores de la función alcanzados por el algoritmo en la función 2 del *benchmark* utilizado. Se puede observar un comportamiento muy similar al problema anterior. También se puede observar que en este caso, al ser el parámetro mayor a 0.05 y acercarse el valor del parámetro de relajación de las distribuciones marginales a 0.1, el valor de la media, de la varianza y de la desviación estándar mejoran en comparación con el caso donde el parámetro de relajación es igual a 1. En este problema se puede ver que cuando el parámetro de relajación es igual a 0.8, la media y el valor máximo son los que tienen menor valor con respecto a los experimentos con los otros valores del parámetro, además de que la solución con valor mínimo obtenida con este mismo valor del parámetro de relajación es la más cercana a la mínima obtenida cuando el parámetro de relajación es igual a 1.

| Parámetro de relajación | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-------------------------|--------------|---------------|---------------|---------------|---------------|--------------|
| Media | 2.9324185042 | 25.7781064516 | 19.7264774194 | 17.1914870968 | 8.7987087097 | 3.3468426129 |
| Desviación Estándar | 1.6560399134 | 3.196496983 | 3.2218639996 | 2.6767327427 | 4.3653228431 | 3.0078252993 |
| Varianza | 2.7424681947 | 10.2175929626 | 10.3804076318 | 7.1648981758 | 19.0560435248 | 9.047016821 |
| Máximo | 6.76754 | 31.1706 | 25.3175 | 24.1458 | 22.4852 | 18.0644 |
| Mínimo | 0.00653336 | 20.4737 | 13.2383 | 11.2893 | 2.48598 | 0.231701 |

| Parámetro de relajación | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Media | 2.6071935484 | 2.7485429065 | 2.387246119 | 3.0287093548 | 2.0329532606 |
| Desviación Estándar | 1.0385574403 | 1.9813149467 | 1.0597248887 | 1.504043733 | 0.9674295012 |
| Varianza | 1.0786015569 | 3.9256089179 | 1.1230168397 | 2.2621475507 | 0.9359198397 |
| Máximo | 4.64742 | 10.0627 | 4.00952 | 9.67419 | 3.1517 |
| Mínimo | 1.02088 | 0.0194001 | 0.00743969 | 1.00698 | 0.00906108 |

Tabla 4.2: Función 2

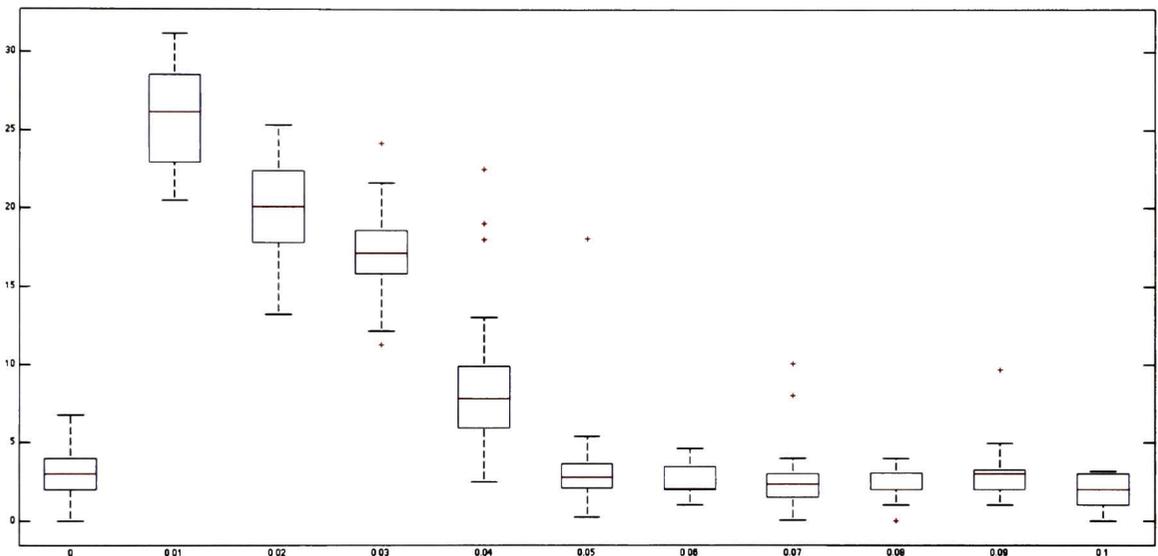


Figura 4.2: Función 2

En la figura 4.2 se muestra un diagrama de caja con los resultados de las 31 corridas para cada valor del parámetro de relajación, optimizando la función 2. En este diagrama de caja se puede apreciar que mediante la utilización de la estrategia de relajación de los parámetros de las distribuciones marginales con valores iguales o mayores a 0.05, los experimentos obtienen típicamente mejores resultados.

En la tabla 4.3 se reportan las estadísticas para los valores de la función alcanzados por el algoritmo en la función 3 del *benchmark* utilizado. Se puede observar que el comportamiento sigue los patrones de los casos anteriores, incluso la mejora es más acentuada con valores grandes del

| Parámetro de relajación | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Media | 0.3415224839 | 4.1798106452 | 0.5576216774 | 0.0854868419 | 0.0597901645 | 0.0492388677 |
| Desviación Estándar | 0.5793060394 | 0.3561295343 | 0.1162456874 | 0.0126588286 | 0.0096261695 | 0.0085230507 |
| Varianza | 0.3355954873 | 0.1268282452 | 0.0135130598 | 0.0001602459 | 9.26631E-005 | 7.26424E-005 |
| Máximo | 2.03277 | 4.99168 | 0.803545 | 0.109446 | 0.0829075 | 0.0636024 |
| Mínimo | 0.0214446 | 3.5217 | 0.347399 | 0.0504266 | 0.0417291 | 0.029396 |

| Parámetro de relajación | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
|-------------------------|--------------|--------------|--------------|--------------|--------------|
| Media | 0.0438245484 | 0.0376476129 | 0.0370533613 | 0.0332866935 | 0.0326840387 |
| Desviación Estándar | 0.008320143 | 0.0064482529 | 0.006409029 | 0.0059665328 | 0.0057232719 |
| Varianza | 6.92248E-005 | 0.00004158 | 4.10756E-005 | 3.55995E-005 | 3.27558E-005 |
| Máximo | 0.059822 | 0.0480462 | 0.0494799 | 0.0458322 | 0.044196 |
| Mínimo | 0.0264074 | 0.0226874 | 0.0154605 | 0.0184878 | 0.0225671 |

Tabla 4.3: Función 3

parámetro de relajación.

En la figura 4.3 se muestra un diagrama de caja con los resultados de las 31 corridas para cada valor del parámetro de relajación, optimizando la función 3. En este diagrama de caja se puede apreciar que mediante la utilización de la estrategia de relajación de los parámetros de las distribuciones marginales con valores iguales o mayores a 0.05, los experimentos obtienen típicamente mejores resultados. La mejora que se presenta es tomando en cuenta que cuando no existe relajación de las distribuciones univariadas se presenta en la experimentación valores atípicos donde las soluciones obtenidas tiene un valor grande, mientras que en los experimentos donde el parámetro de relajación de las distribuciones univariadas es igual o mayor a 0.05, los valores máximos obtenidos son menores al máximo típico de los experimentos donde la relajación de parámetros no existe, además de no presentarse valores atípicos cuando el parámetro de relajación de las distribuciones univariadas es igual o mayor a 0.05, y cuando se presentan valores atípicos, estos son menores a los valores típicos del mínimo en el conjunto de experimentos.

En la tabla 4.4 se reportan las estadísticas para los valores de la función alcanzados por el algoritmo en la función 19 del *benchmark* utilizado. En general, los valores en las estadísticas de los experimentos cuando se utiliza la relajación de parámetros disminuyen o mejoran en comparación cuando no se utiliza la estrategia de relajación de parámetros, teniendo una tendencia a disminuir cada vez más cuando el parámetro de relajación se acerca a 0.08.

En la figura 4.4 se muestra un diagrama de caja con los resultados de las 31 corridas para cada valor del parámetro de relajación, optimizando la función 19. Este diagrama se comprueban los

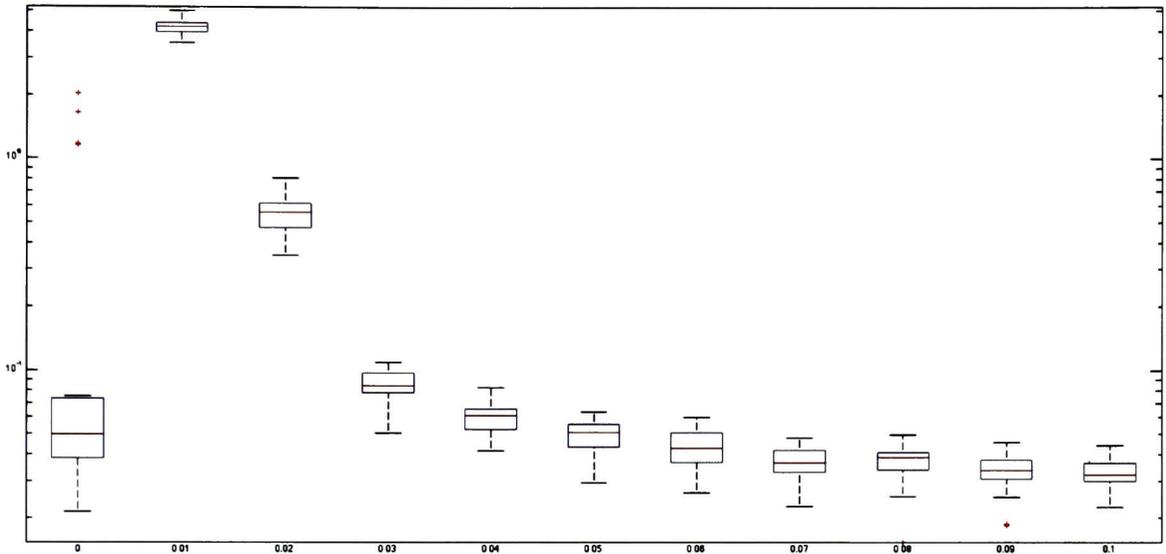


Figura 4.3: Función 3

| Parámetro de relajación | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-------------------------|------------------|------------------|-----------------|----------------|----------------|----------------|
| Media | 698.3242612903 | 648.7506451613 | 114.9755645161 | 44.682866129 | 27.9047125806 | 19.9983444194 |
| Desviación Estándar | 505.066190287 | 163.1927922797 | 50.9340345053 | 31.3399185987 | 23.1276266694 | 19.1062371871 |
| Varianza | 255091.856571067 | 26631.8874520366 | 2594.2758709877 | 982.1904977746 | 534.8871153612 | 365.0482994479 |
| Máximo | 1873.6 | 1096.97 | 224.814 | 130.053 | 113.697 | 85.8136 |
| Mínimo | 25.485 | 321.8 | 19.7118 | 4.90284 | 2.01571 | 0.832647 |

| Parámetro de relajación | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
|-------------------------|----------------|----------------|---------------|----------------|----------------|
| Media | 14.733619871 | 22.3287542871 | 11.3200386452 | 13.6700926774 | 17.7161984516 |
| Desviación Estándar | 18.5133477221 | 22.8224320997 | 8.7908195136 | 12.6903678199 | 17.751522442 |
| Varianza | 342.7440438805 | 520.8634069434 | 77.2785077216 | 161.0454354055 | 315.1165490081 |
| Máximo | 75.9221 | 113.528 | 26.5464 | 56.7822 | 75.7428 |
| Mínimo | 0.6111797 | 0.0865839 | 0.154871 | 0.633953 | 0.333697 |

Tabla 4.4: Función 19

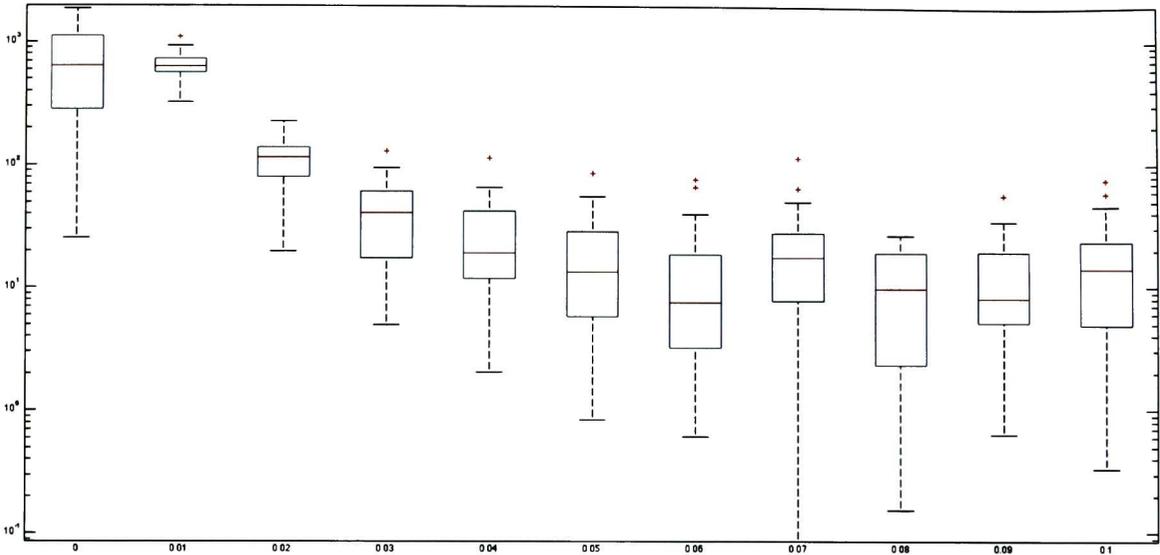


Figura 4.4: Función 19

comentarios anteriores.

En la tabla 4.5 se reportan las estadísticas para los valores de la función alcanzados por el algoritmo en la función 20 del *benchmark* utilizado. Nuevamente, los resultados siguen la misma tendencia, y mejora incrementando el parámetro de relajación.

En la figura 4.5 se muestra un diagrama de caja con los resultados de las 31 corridas para cada valor del parámetro de relajación, optimizando la función 20.

| Parámetro de relajación | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-------------------------|------------------|------------------|-----------------|---------------|---------------|---------------|
| Media | 7.40E+005 | 26862.3690322581 | 97.7289709677 | 11.6189977419 | 10.4979051613 | 10.7321106452 |
| Desviación Estándar | 2670205.32289216 | 10892.1393355023 | 32.2577641908 | 0.9598876798 | 0.6866521235 | 1.0981190537 |
| Varianza | 7129996466401.61 | 118638699.303996 | 1040.5633505875 | 0.9213843578 | 0.4714911387 | 1.2058654562 |
| Máximo | 1.28E+007 | 49229.1 | 178.008 | 13.4161 | 11.6509 | 15.7415 |
| Mínimo | 9.92 | 9689.04 | 52.7974 | 9.59041 | 8.7676 | 9.13625 |

| Parámetro de relajación | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
|-------------------------|---------------|---------------|--------------|---------------|---------------|
| Media | 10.5699983871 | 10.4426612903 | 10.252466129 | 10.1902796774 | 10.1434487097 |
| Desviación Estándar | 0.4663213953 | 0.622419765 | 0.5365147617 | 0.398000088 | 0.5049662524 |
| Varianza | 0.2174556437 | 0.3874063638 | 0.2878480895 | 0.15840407 | 0.254990916 |
| Máximo | 11.6292 | 11.7466 | 11.5262 | 11.133 | 10.9838 |
| Mínimo | 9.54426 | 9.18804 | 9.25793 | 9.54038 | 8.92988 |

Tabla 4.5: Función 20

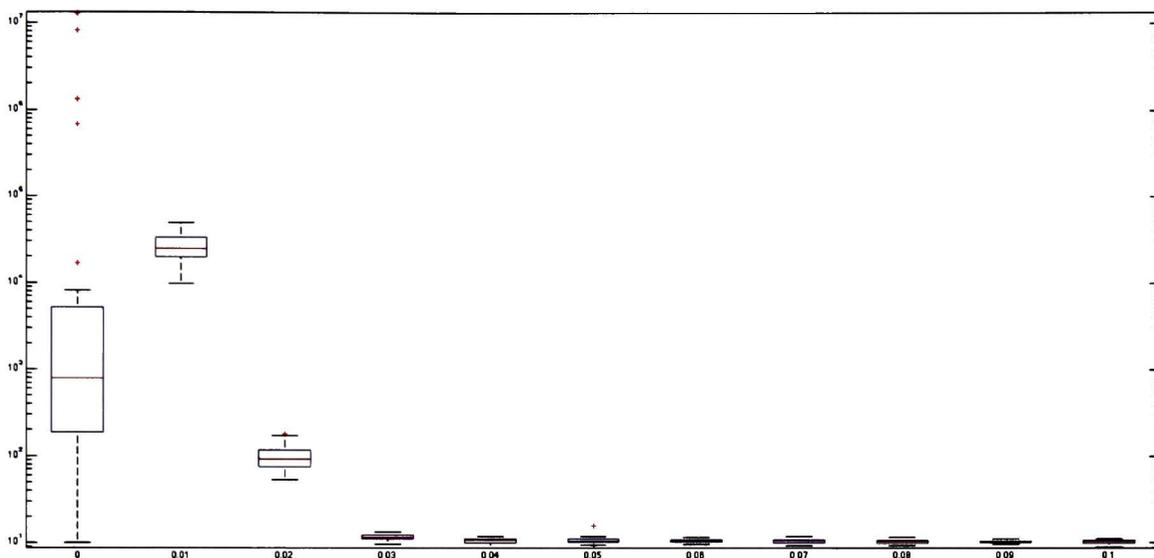


Figura 4.5: Función 20

4.4 Experimentos utilizando el calculo automático del parámetro de relajación

Los experimentos se llevaron a cabo sobre las 20 funciones de las que se compone el *benchmark* utilizado con un total de 31 corridas por problema.

Los parámetros utilizados en la experimentación son:

- Población: 100 individuos
- Número máximo de generaciones: 1000, 2500, 5000 y 25000 (máximo 100000, 250000, 500000 y 2500000 evaluaciones)
- Número máximo de padres: 5 (lo que supone cópulas con un total de 6 variables)
- Porcentaje de selección: 30
- Dimensionalidad de los problemas: 10, 50, 100 y 500 variables de decisión

| Problema | 1 | 2 | 3 |
|---------------------|-------------|--------------|-------------|
| Mejor | 2.8901E-016 | 0 | 8.2308E-009 |
| Mediana | 7.3788E-016 | 2.98488 | 9.6216E-009 |
| Peor | 1.0178E-015 | 22.6914 | 1.1723E-008 |
| Promedio | 7.0108E-016 | 6.736441871 | 9.8832E-009 |
| Desviación estándar | 1.6830E-016 | 6.7769123905 | 9.9152E-010 |

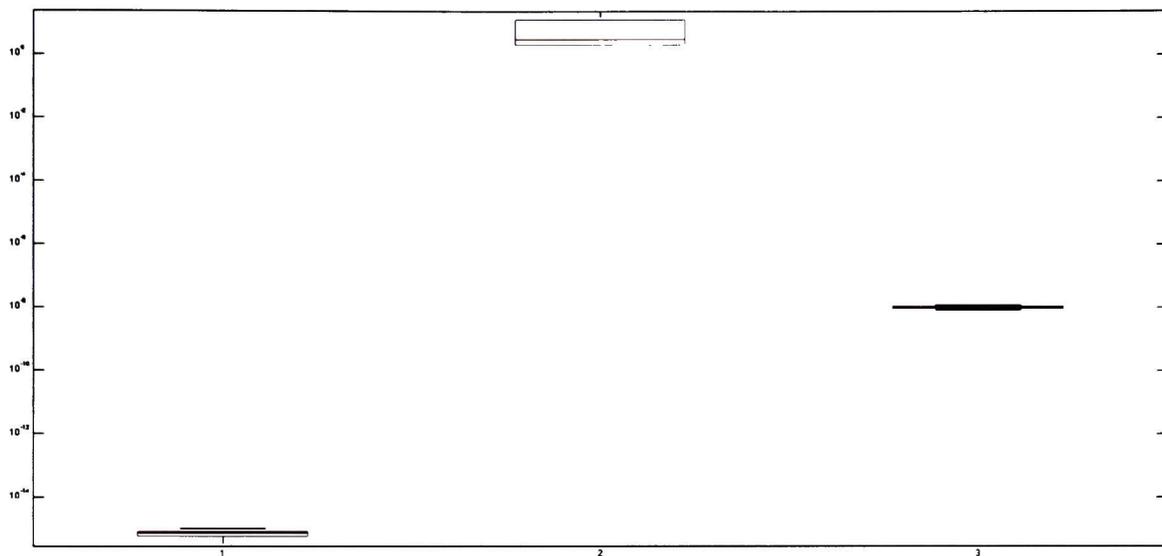
Tabla 4.6: Funciones separables

4.4.1 Experimentos con 10 variables

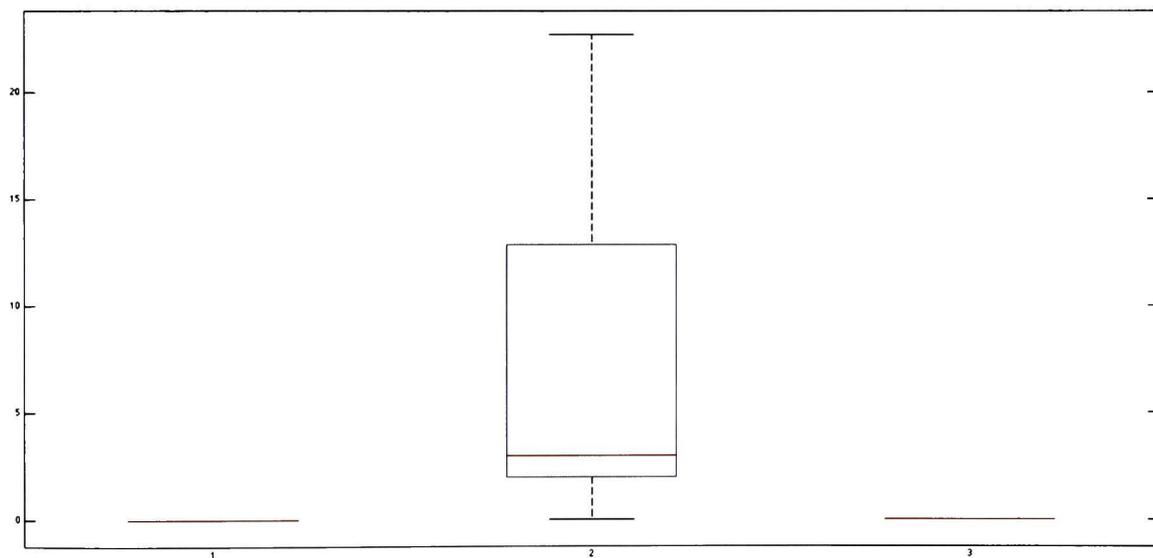
En la tabla 4.6 se reportan las estadísticas para los valores alcanzados por el algoritmo en las funciones separables del *benchmark* utilizado. Se puede observar que en las tres funciones se obtienen valores muy cercanos al óptimo global, inclusive en la función 2 se obtiene el óptimo global. Sin embargo, de la desviación estándar también puede verse que los resultados del problema 2 son los menos estables, teniendo una alta variabilidad. En los otros casos las desviación estándar es relativamente pequeña. Cabe resaltar que la función en la cual se obtienen mejores resultados es la unimodal.

En la figura 4.6 se muestran dos diagramas de caja con los resultados de las 31 corridas para cada función dentro del grupo de las funciones separables. En estos diagramas de caja se puede apreciar la buena calidad y poca variabilidad de los resultados obtenidos en las funciones 1 y 3. Además, se puede observar que para la función 2, los casos donde se obtiene el óptimo de la función son extremos y la mayoría de los resultados obtenidos aunque no son muy lejanos del óptimo global, no tienen una calidad aceptable.

En la tabla 4.7 se reportan las estadísticas para los valores alcanzados por el algoritmo en las funciones con un grupo no separable de tamaño m del *benchmark* utilizado. Se puede observar que el algoritmo se comporta mejor en la función 7, en la cual se obtienen valores pequeños en todos sus indicadores. El buen comportamiento en esta función puede atribuirse a que es unimodal. La función 6 obtiene mayormente resultados buenos, aun tratándose de una función multimodal. Se puede ver que la varianza tiene valor relativamente mediano, seguramente por la influencia del peor valor, mismo que sin embargo parece ser un resultado atípico. La función 4 es la que peores resultados



(a) Escala logarítmica



(b) Escala lineal

Figura 4.6: Funciones separables

| Problema | 4 | 5 | 6 | 7 | 8 |
|---------------------|-------------|-------------|--------------|-------------|---|
| Mejor | 1.5594E+008 | 0.994959 | 0.00394539 | 1.5530E-011 | - |
| Mediana | 3.4003E+010 | 26.6121 | 0.00588351 | 9.2561E-011 | - |
| Peor | 2.0146E+011 | 5027870 | 19.2532 | 2.5940E-010 | - |
| Promedio | 4.3875E+010 | 7.2053E+005 | 1.8226166694 | 1.1148E-010 | - |
| Desviación estándar | 4.7600E+010 | 1.0464E+006 | 5.6431767084 | 6.1453E-011 | |

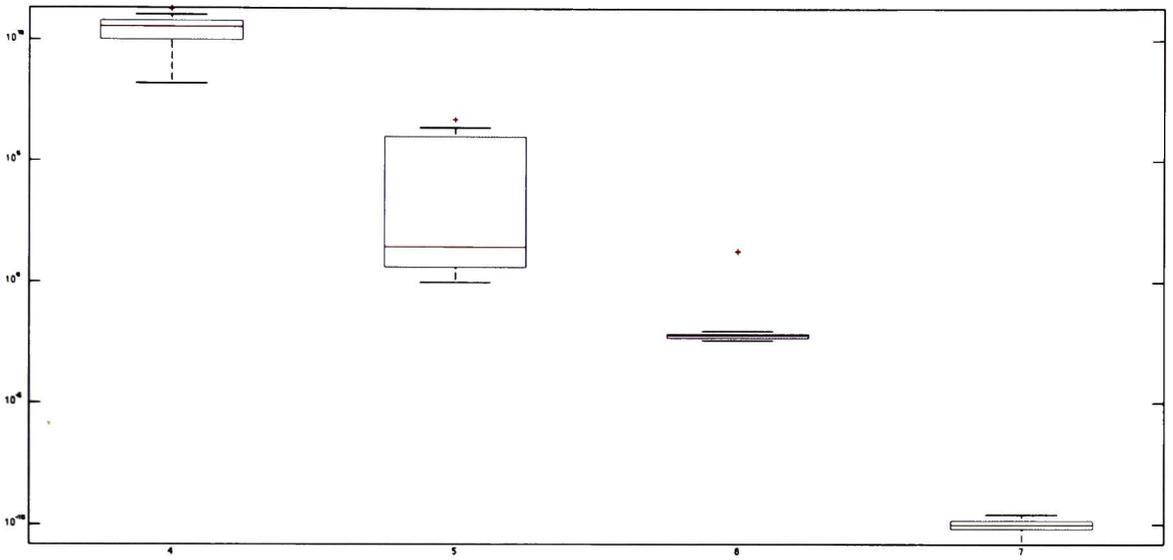
Tabla 4.7: Funciones con un grupo no separable de tamaño m

obtiene, aun siendo unimodal, por causas que permanecen pendientes de investigar.

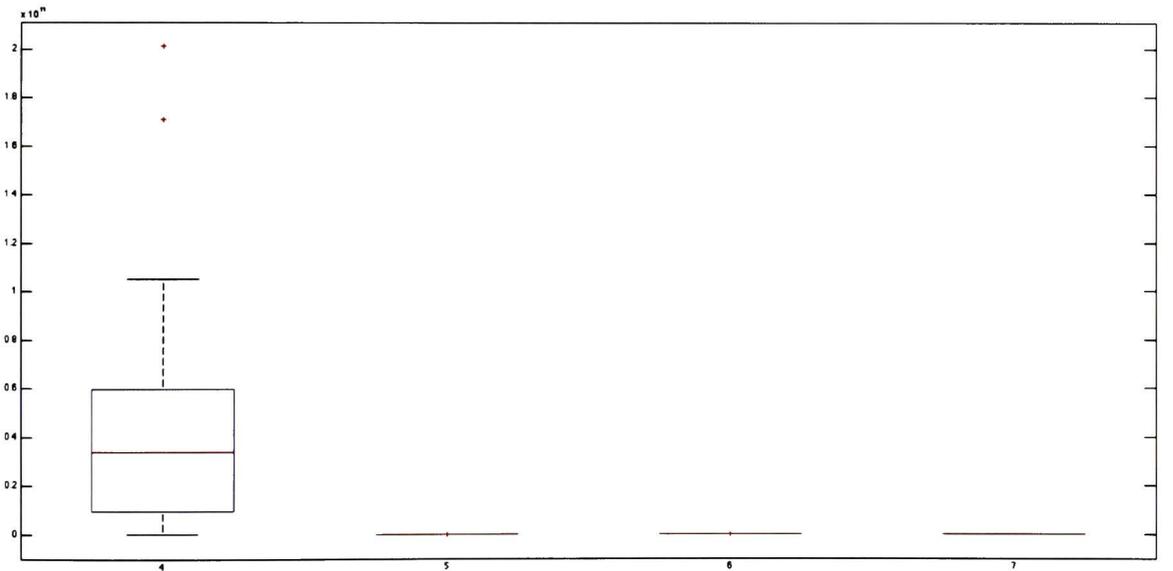
En la figura 4.7 se muestran dos diagramas de caja con los resultados de las 31 corridas para cada función dentro del grupo de las funciones con un grupo no separable de tamaño m . En estos diagramas de caja se puede verificar el buen comportamiento de las funciones 6 y 7. Se puede corroborar que el valor de la varianza de los resultados de la función 6 se ve afectada por valores atípicos. Para las funciones 4 y 5 se puede apreciar que los resultados no son muy buenos.

En la tabla 4.8 se reportan las estadísticas para los valores alcanzados por el algoritmo en las funciones con $\frac{D}{2m}$ grupos no separables de tamaño m del *benchmark* utilizado. Se puede observar que en las funciones 11 y 12, multimodal y unimodal respectivamente, se obtienen resultados cercanos al óptimo en todas las corridas con muy pequeña variación entre los resultados obtenidos entre las corridas. En la función 13 se obtienen valores aceptables, igualmente con poca variación entre los resultados obtenidos en las corridas. En la función 10 se obtuvo una mejor solución de buena calidad, sin embargo los valores de la media, promedio, peor solución y desviación estándar sugieren que los resultados en general no son tan buenos. La función en la que peores resultados se obtuvieron fue la numero 9, donde la mejor solución esta lejos del óptimo global, y la desviación estándar es muy grande, nuevamente tratándose de una función unimodal. La(s) característica(s) que presenta(n) dificultades para la técnica no lograron determinarse en este estudio, y requieren de análisis mas profundo.

En la figura 4.8 se muestran dos diagramas de caja con los resultados de las 31 corridas para cada función dentro del grupo de las funciones con $\frac{D}{2m}$ grupos no separables de tamaño m . Mediante estos diagramas de caja se puede corroborar el comportamiento descrito en el párrafo anterior de acuerdo a la tabla 4.8. Para la función 10 se puede confirmar que la mayoría de los resultados se



(a) Escala logarítmica



(b) Escala lineal

Figura 4.7: Funciones con un grupo no separable de tamaño m

| Problema | 9 | 10 | 11 | 12 | 13 |
|---------------------|------------|-------------|-------------|-------------|-------------|
| Mejor | 3178.21 | 4.0728E-011 | 1.4841E-008 | 4.7017E-016 | 0.554944 |
| Mediana | 90349.4 | 12.2665 | 1.9219E-008 | 8.1365E-016 | 0.84395 |
| Peor | 240071 | 20.2188 | 2.2940E-008 | 1.1986E-015 | 1.06693 |
| Promedio | 98674.5439 | 11.22233516 | 1.8887E-008 | 7.9668E-016 | 0.862956161 |
| Desviación estándar | 70014.9232 | 6.599827012 | 2.3388E-009 | 2.1094E-016 | 0.121421215 |

Tabla 4.8: Funciones con $\frac{D}{2m}$ grupos no separables de tamaño m

| Problema | 14 | 15 | 16 | 17 | 18 |
|---------------------|-------------|-------------|-------------|-------------|--------------|
| Mejor | 11093.2 | 0.994959 | 1.6205E-008 | 2.5703E-015 | 0.0302061 |
| Mediana | 205076 | 17.4242 | 1.9274E-008 | 4.2062E-010 | 0.0962239 |
| Peor | 657457 | 24.4722 | 2.3370E-008 | 8.3823E-005 | 12.8176 |
| Promedio | 253148.9226 | 15.36540510 | 1.9416E-008 | 4.0698E-006 | 0.6185593581 |
| Desviación estándar | 149257.3308 | 6.491048570 | 1.5705E-009 | 1.5966E-005 | 2.289251686 |

Tabla 4.9: Funciones con $\frac{D}{m}$ grupos no separables de tamaño m

alejan de las buenas regiones, y la mejor solución obtenida se trata de un valor extremo.

En la tabla 4.9 se reportan las estadísticas para los valores alcanzados por el algoritmo en las funciones con $\frac{D}{m}$ grupos no separables de tamaño m del *benchmark* utilizado. Se puede observar un comportamiento muy parecido al grupo anterior de funciones.

En la figura 4.9 se muestran dos diagramas de caja con los resultados de las 31 corridas para cada función dentro del grupo de las funciones con $\frac{D}{m}$ grupos no separables de tamaño m . En estos diagramas de caja se pueden apreciar los buenos resultados de las funciones 16, 17 y 18. Aunque los resultados obtenidos en la función 17 tienen una variación notable, incluso los valores atípicos son de buena calidad. En lo que respecta a la función 18 se puede apreciar que la mayoría de los resultados son de buena calidad, siendo algunos valores atípicos los que hacen que la totalidad de las estadísticas obtenidas no sean tan buenas. Para la función 15 se puede observar que los mejores resultados que pudieron obtenerse se tratan de valores atípicos.

En la tabla 4.10 se reportan las estadísticas para los valores alcanzados por el algoritmo en las funciones no separables del *benchmark* utilizado. Se puede observar que la función 19, la cual es unimodal presenta buena calidad de los resultados obtenidos y pequeño valor de la desviación

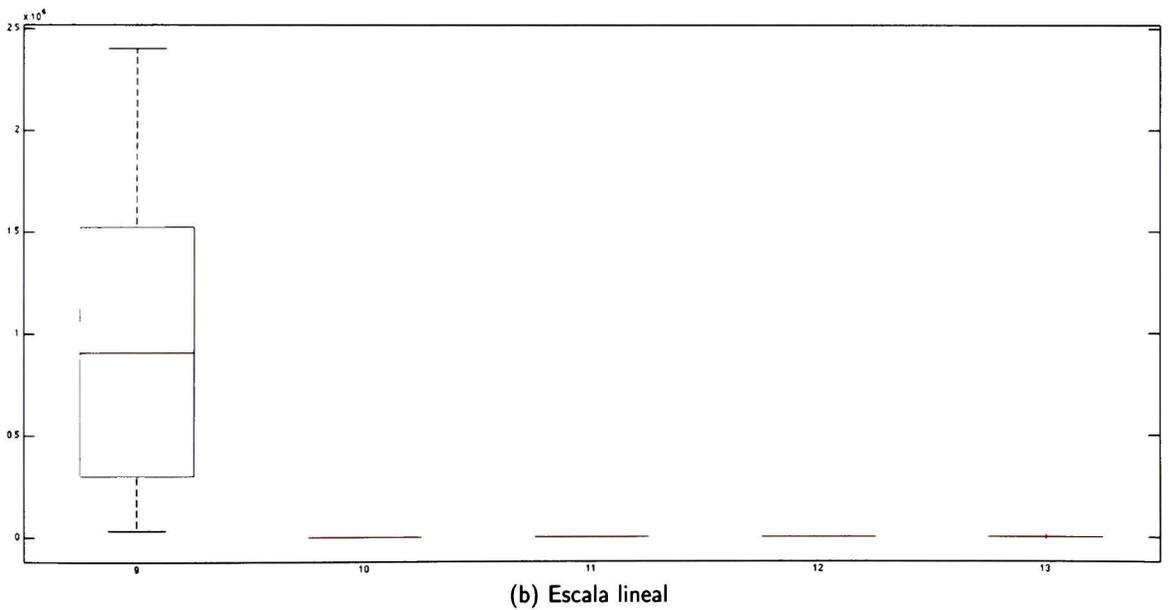
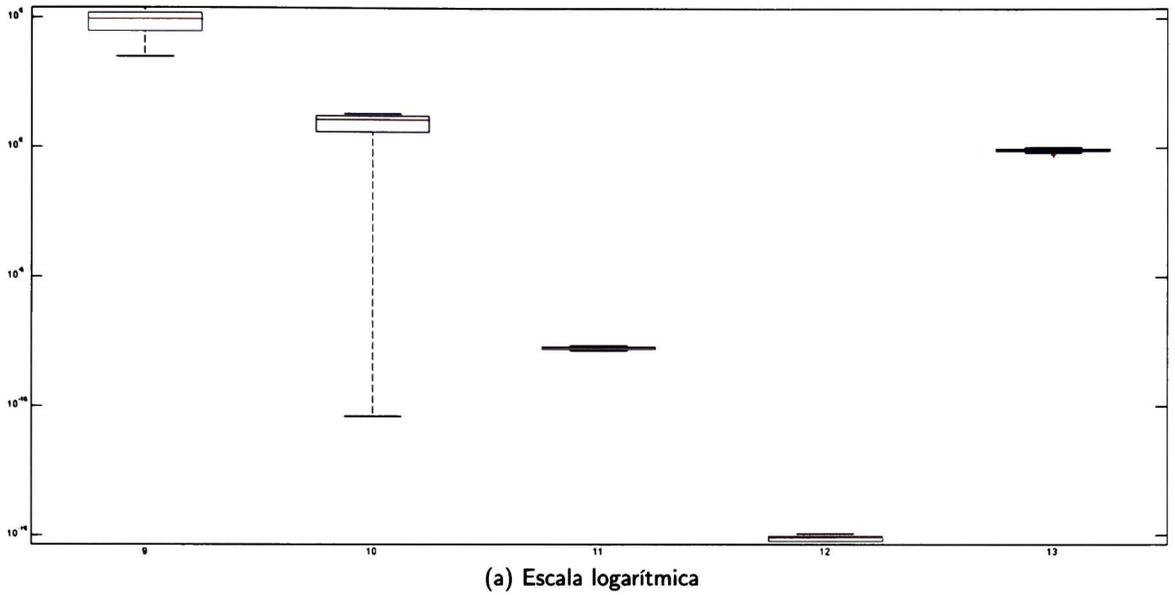
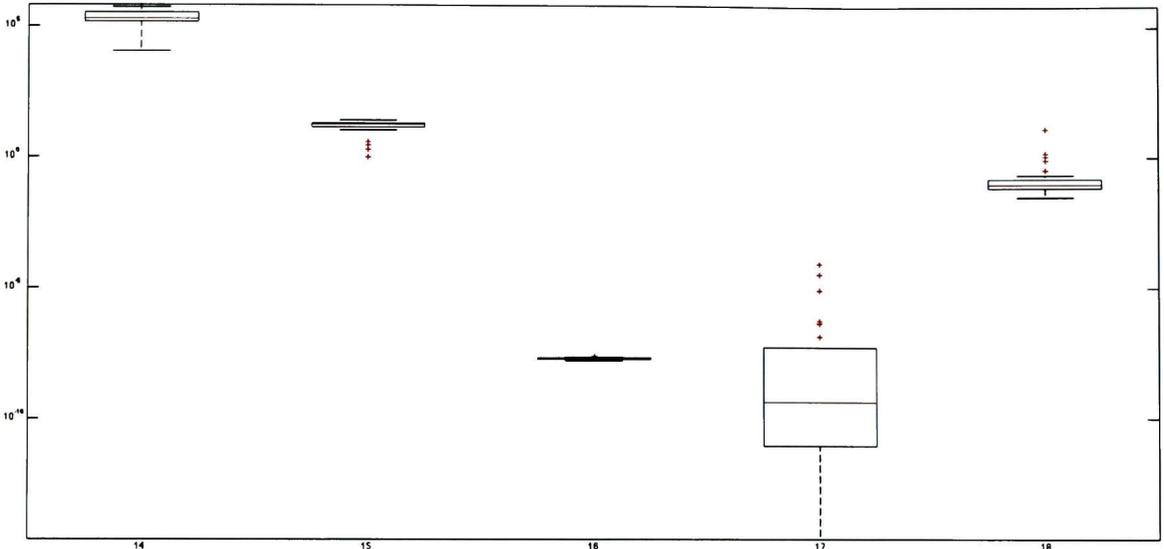
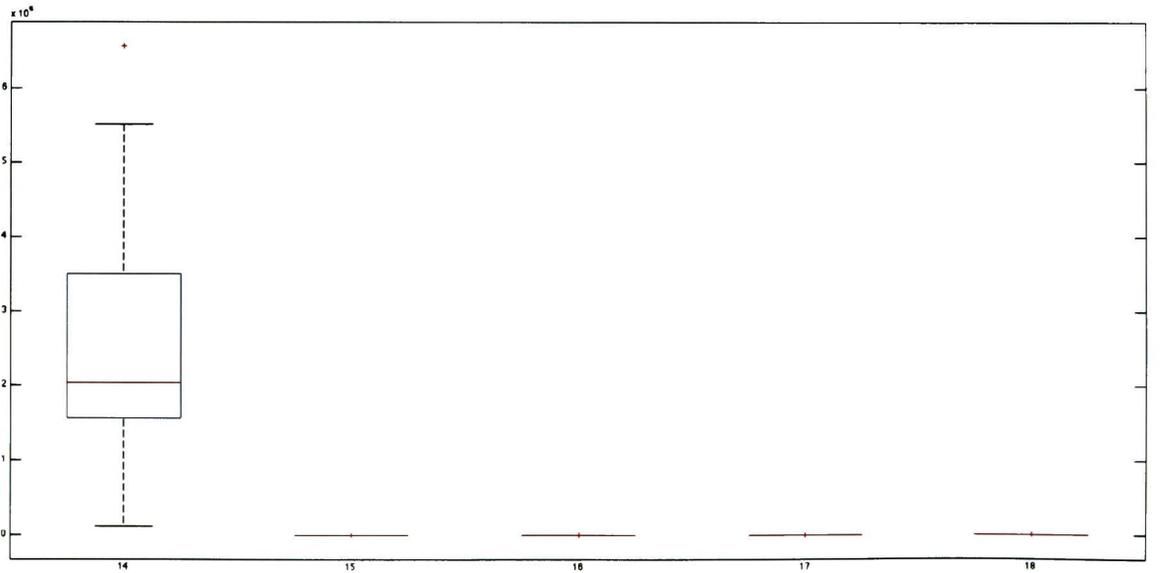


Figura 4.8: Funciones con $\frac{D}{2m}$ grupos no separables de tamaño m



(a) Escala logarítmica



(b) Escala lineal

Figura 4.9: Funciones con $\frac{D}{m}$ grupos no separables de tamaño m

| Problema | 19 | 20 |
|---------------------|-------------|-------------|
| Mejor | 8.6581E-013 | 2.75296 |
| Mediana | 1.0155E-006 | 4.9144 |
| Peor | 3.9616E-003 | 5.69343 |
| Promedio | 2.6346E-004 | 4.817186129 |
| Desviación estándar | 8.9630E-004 | 0.693259345 |

Tabla 4.10: Funciones no separables

| Problema | 7 |
|---------------------|---------------|
| Mejor | 0.000822649 |
| Mediana | 3.69796 |
| Peor | 56.3223 |
| Promedio | 8.0067211984 |
| Desviación estándar | 11.8115884029 |

Tabla 4.11: Funciones con un grupo no separable de tamaño m

estándar. El comportamiento del algoritmo con la función 20 es muy estable arrojando resultados con poca variación. Este último problema es multimodal, lo que afecta levemente a la calidad de los resultados.

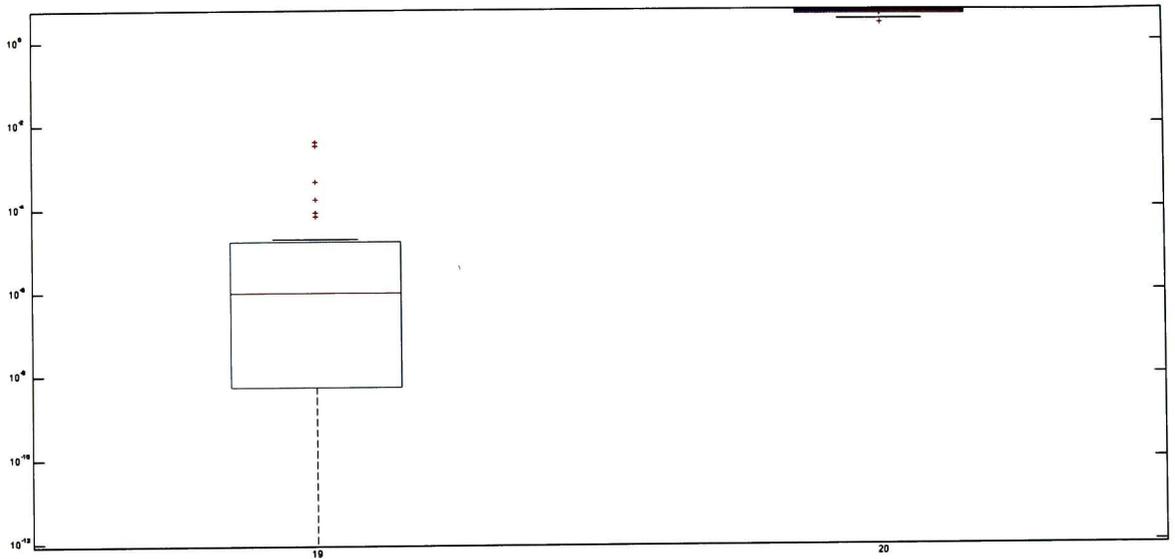
En la figura 4.9 se muestran dos diagramas de caja con los resultados de las 31 corridas para cada función dentro del grupo de las funciones no separables. Mediante estos diagramas de caja se puede corroborar el comportamiento descrito en el párrafo anterior de acuerdo a la tabla 4.10.

4.4.2 Experimentos con 50 variables

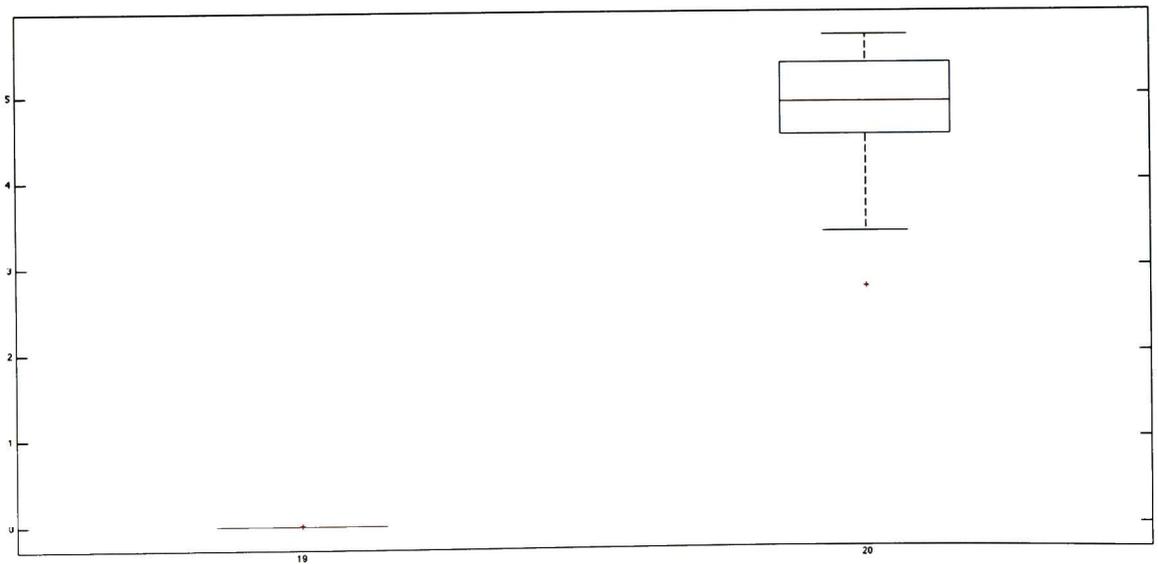
En esta sección se presentan algunos resultados debido al alto costo computacional que observa el algoritmo propuesto. Es por eso que se reportan los resultados obtenidos de las modificaciones del problema 1.2 de Schwefel.

En la tabla 4.11 se reportan las estadísticas para los valores alcanzados por el algoritmo en la función 7, que pertenece al grupo de funciones con un grupo no separable de tamaño m . Se puede observar la obtención de resultados de calidad aceptable, donde el valor de la desviación estándar es aparentemente afectado por valores atípicos.

En la tabla 4.12 se reportan las estadísticas para los valores alcanzados por el algoritmo en la



(a) Escala logarítmica



(b) Escala lineal

Figura 4.10: Funciones no separables

| | |
|---------------------|----------------|
| Problema | 12 |
| Mejor | 31.3256 |
| Mediana | 130.243 |
| Peor | 350.252 |
| Promedio | 144.3651516129 |
| Desviación estándar | 89.0408845532 |

Tabla 4.12: Funciones con $\frac{D}{2m}$ grupos no separables de tamaño m

| | |
|---------------------|-----------------|
| Problema | 17 |
| Mejor | 5921.59 |
| Mediana | 9645.27 |
| Peor | 16715.3 |
| Promedio | 9987.3922580645 |
| Desviación estándar | 2996.3154913479 |

Tabla 4.13: Funciones con $\frac{D}{m}$ grupos no separables de tamaño m

función 12, que pertenece al grupo de funciones con $\frac{D}{2m}$ grupos no separables de tamaño m .

En la tabla 4.13 se reportan las estadísticas para los valores alcanzados por el algoritmo en la función 17, que pertenece al grupo de funciones con $\frac{D}{m}$ grupos no separables de tamaño m .

En la tabla 4.14 se reportan las estadísticas para los valores alcanzados por el algoritmo en la función 19, que pertenece al grupo de funciones no separables.

En estos cuatro casos se puede observar que conforme se hacen mas complejas las dependencias entre variables, se obtienen resultados de menor calidad si el número de variables crece.

Es decir, el impacto de las relaciones complejas entre variables es más notable en cuanto se incrementa el número de variables.

| | |
|---------------------|------------------|
| Problema | 19 |
| Mejor | 6364.14 |
| Mediana | 9438.25 |
| Peor | 17547.3 |
| Promedio | 10212.3561290323 |
| Desviación estándar | 2848.5903412538 |

Tabla 4.14: Funciones no separables

Cabe aclarar que el número de experimentos reportados se vió limitado debido al importante costo computacional asociado a las ejecuciones del algoritmo. Con este hecho comprobamos que aunque los algoritmos de estimación de distribución son excelentes modelos matemáticos para comprender los algoritmos evolutivos, estos últimos siguen siendo más eficientes en la práctica.

4.5 Resumen

En este capítulo se presentaron los detalles de la implementación del algoritmo, el cual fue implementado en lenguaje C++ para hacer uso del modelado orientado a objetos. Las cópulas arquimedeanas de un parámetro se modelaron mediante una clase con los atributos inherentes a éstas, y cada una de las cópulas utilizadas se modelo utilizando esta clase como base y desarrollando las operaciones específicas de cada cópula.

También se presentaron los experimentos llevados a cabo, los cuales utilizaron las funciones propuestas en [Tang et al., 2010] que esta compuesto de 6 funciones base con diferentes características, las cuales se modifican para obtener diferentes grados de complejidad en las relaciones presentes entre las variables de cada función.

Inicialmente se probó con un conjunto de 5 funciones representativas con características diferentes utilizando el parámetro de relajación fijo. Estos experimentos mostraron que la propuesta era promisoria y sirvieron para darnos una idea de los valores con los que el parámetro de relajación causaba que el algoritmo obtuviera mejores resultados. Esto sirvió como base para proponer una regla heurística para el cálculo automático del parámetro de relajación.

Una vez que se obtuvo una regla heurística para el cálculo del parámetro de relajación, se realizaron experimentos en todas las funciones del *benchmark* utilizado, en los cuales se pudo observar el buen comportamiento en ciertas funciones que incluyen diferentes grados de complejidad en sus relaciones y diferente forma en la superficie del espacio del búsqueda.

Un número limitado de experimentos se reportaron en dimensiones mayores debido al alto costo computacional asociado a las ejecuciones del algoritmo.

5

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones obtenidas a partir de la elaboración de este trabajo de tesis. De igual manera se dan a conocer así como algunos puntos considerados como trabajo a futuro para la mejora del trabajo.

5.1 Conclusiones

Los algoritmos de estimación de distribución es un reciente enfoque de optimización dentro del computo evolutivo, el cual ha probado ser de gran utilidad, debido a la rápida convergencia que estos presentan. Sin embargo, esta propiedad puede ser vista como una desventaja cuando este tipo de algoritmos se enfrentan a problemas que presentan multimodalidades u otras características adversas. Este comportamiento se da ya que al evolucionar las poblaciones de individuos a través de las generaciones, los mejores individuos en cada generación, a partir de los cuales se induce el modelo probabilístico que generara nuevos individuos para la siguiente generación, se encontraran cada vez mas cercanos entre si, lo que dará como resultado que la mayor parte de los individuos de la siguiente generación sean muestreados en esta área. Tal comportamiento puede llevar a la

convergencia prematura.

La relajación de las distribuciones marginales mediante las reglas utilizadas en este trabajo evita este comportamiento al permitir que los parámetros de las generaciones anteriores, en las cuales la probabilidad mayor estará distribuida en un área más grande del espacio de búsqueda que el de la generación actual (debido al comportamiento antes descrito), influyen los parámetros de las distribuciones aprendidos en la generación actual. Esto se pudo comprobar mediante la medición de la varianza poblacional en cada generación, pues al tomar en cuenta solo los parámetros de la generación actual esta decrecía a un velocidad muy grande, haciendo que el algoritmo quedara estancado en cierta área del espacio de búsqueda, la cual en muchos casos era la de un óptimo local. En cambio, al aplicar la relajación de distribuciones, esta varianza se veía reducida en menor proporción. Es entonces que podemos concluir que mediante la aplicación del enfoque de relajación de las distribuciones, se permitió una mayor exploración del espacio de búsqueda, permitiendo escapar del estancamiento del algoritmo en óptimos locales.

Sin embargo a pesar de lo antes mencionado, debido a que la relajación de los parámetros de las distribuciones marginales se da mediante un parámetro, el valor que tome este parámetro afectará en qué grado los parámetros de distribución en las generaciones anteriores a los parámetros de distribución en la generación actual. Es por eso que un valor muy pequeño de este puede resultar en una exploración muy extensa del espacio de búsqueda, lo que supone que mientras se siga explorando el espacio de esta manera las soluciones obtenidas no tendrán un refinamiento suficiente para obtener soluciones de buena calidad. En el caso contrario, un valor muy grande del parámetro significaría una influencia menor o casi nula de los parámetros en las generaciones anteriores, lo que eliminaría la bondad que brinda el enfoque de relajación de distribuciones, que es la mayor exploración del espacio de búsqueda para no quedar estancado en óptimos locales.

Alternativamente al enfoque de relajación de las distribuciones, se probó un enfoque donde para poder mantener una diversidad, se seleccionan además de los mejores individuos de una población, un porcentaje menor de los peores individuos de la población. Lo cual no rindió mejores resultados, esto debido a que además de aprender las relaciones y distribuciones de los mejores individuos, también se hacía para los peores individuos, lo que venía a decrementar la calidad del modelo probabilístico.

Además, se pudo comprobar mediante experimentación que la utilización de los 3 tipos de cópulas permite una diversidad en la forma de dependencias entre variables de decisión que pueden ser representadas por el modelo probabilístico, lo que produce mejores resultados que si sólo se utilizara una de las cópulas. En ese caso, por la restricción de que todas las dependencias tendrán la misma forma de dependencia, el modelo probabilístico degrada la forma en que se representan las verdaderas dependencias entre las variables.

Un punto importante que se debe mencionar es el inherente costo computacional que conlleva el proceso de inducir un modelo y muestrear individuos a partir de este, especialmente cuando las distribuciones no tienen formulas cerradas o métodos para encontrar los valores de las funciones de forma directa. En nuestro caso, la utilización de la distribución beta significa un costo computacional notable debido a que la función cuantil debe ser estimada mediante un algoritmo iterativo, y en cuanto más precisión se desee y la probabilidad se concentre en una región mas pequeña, mas tardado sera el proceso de muestreo.

5.2 Trabajo futuro

Una posible ruta de trabajo futuro es obtener una heurística que obtenga un parámetro de relajación de los parámetros de las distribuciones marginales y posiblemente cambie su valor durante el funcionamiento del algoritmo, esto para eliminar el problema de proporcionar este parámetro inicialmente y que esto signifique el decremento calidad de las soluciones obtenidas. La varianza de los individuos a partir de los cuales se induce el modelo probabilístico, puede contener información útil para el calculo inicial y del ajuste del valor del parámetro de relajación a través de la ejecución del algoritmo. Además de esto, la variación de la aptitud entre generaciones puede ser un factor que motive el recálculo o cambio de valor del parámetro de relajación de las distribuciones.

Otra idea relevante para investigación es buscar estrategias para representar las dependencias negativas, las cuales hasta ahora no pueden ser representadas debido a la restricción del modelo probabilístico de sólo tener densidades positivas.

Otra ruta de trabajo futuro relevante es la aplicación de los algoritmos de estimación de

distribución como operador dentro de otros algoritmos, debido a su rápida convergencia natural.

Finalmente, la principal línea de investigación es el diseño de estrategias para que el algoritmo pueda ser escalable a mayores dimensionalidades, ya que aunque el uso del tau de Kendall para agilizar el aprendizaje de la estructura del modelo probabilístico supone menor cantidad de tiempo de cómputo (lo que se cumpliría aun en altas dimensionalidades en comparación con la ausencia de esta agilización del aprendizaje mediante el tau de Kendall), el aumento del número de dimensiones del problema a optimizar puede impactar de manera importante el tiempo de cómputo, dada la necesidad del cálculo de la matriz del tau de Kendall, mismo que requiere un tiempo cuadrático en términos del número de variables de decisión del problema de optimización.

A

Anexo

A continuación se presenta la definición y las propiedades de las funciones utilizadas para experimentación.

Función 1. Función elíptica desplazada

$$F_1(\mathbf{x}) = F_{eliptica}(\mathbf{z}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} z_i^2$$

D = dimensiones

$\mathbf{x} = (x_1, x_2, \dots, x_D)$: la solución candidata - un vector fila D -dimensional

$\mathbf{o} = (o_1, o_2, \dots, o_D)$: el óptimo global desplazado

$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \dots, z_D)$: la solución candidata desplazada - un vector fila D -dimensional

Propiedades:

1. Unimodal
2. Desplazada

3. Separable

4. Escalable

5. $\mathbf{x} \in [-100, 100]^D$

6. Optimo global: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = 0$

Función 2. Función de Rastrigin desplazada

$$F_2(\mathbf{x}) = F_{\text{Rastrigin}}(\mathbf{z}) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10]$$

D = dimensiones

$\mathbf{x} = (x_1, x_2, \dots, x_D)$: la solución candidata - un vector fila D -dimensional

$\mathbf{o} = (o_1, o_2, \dots, o_D)$: el optimo global desplazado

$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \dots, z_D)$: la solución candidata desplazada - un vector fila D -dimensional

Propiedades:

1. Multimodal

2. Desplazada

3. Separable

4. Escalable

5. $\mathbf{x} \in [-5, 5]^D$

6. Optimo global: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = 0$

Función 3. Función de Ackley desplazada

$$F_3(\mathbf{x}) = F_{ackley}(\mathbf{z}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e$$

D = dimensiones

$\mathbf{x} = (x_1, x_2, \dots, x_D)$: la solución candidata - un vector fila D -dimensional

$\mathbf{o} = (o_1, o_2, \dots, o_D)$: el óptimo global desplazado

$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \dots, z_D)$: la solución candidata desplazada - un vector fila D -dimensional

Propiedades:

1. Multimodal
2. Desplazada
3. Separable
4. Escalable
5. $\mathbf{x} \in [-32, 32]^D$
6. Óptimo global: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = 0$

Función 19. Problema 1.2 de Schwefel desplazado

$$F_{19}(\mathbf{x}) = F_{schwefel}(\mathbf{z}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$$

D = dimensiones

$\mathbf{x} = (x_1, x_2, \dots, x_D)$: la solución candidata - un vector fila D -dimensional

$\mathbf{o} = (o_1, o_2, \dots, o_D)$: el óptimo global desplazado

$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \dots, z_D)$: la solución candidata desplazada - un vector fila D -dimensional

Propiedades:

1. Unimodal
2. Desplazada
3. No separable
4. $\mathbf{x} \in [-100, 100]^D$
5. Optimo global: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = 0$

Función 20. Función de Rosenbrock desplazada

$$F_{20}(\mathbf{x}) = F_{\text{rosenbrock}}(\mathbf{z}) = \sum_{i=1}^{D-1} \left[100 (z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right]$$

D = dimensiones

$\mathbf{x} = (x_1, x_2, \dots, x_D)$: la solución candidata - un vector fila D -dimensional

$\mathbf{o} = (o_1, o_2, \dots, o_D)$: el optimo global desplazado

$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \dots, z_D)$: la solución candidata desplazada - un vector fila D -dimensional

Propiedades:

1. Multimodal
2. Desplazada
3. No separable
4. $\mathbf{x} \in [-100, 100]^D$
5. Optimo global: $\mathbf{x}^* = \mathbf{o} + 1$, $F_1(\mathbf{x}^*) = 0$

Bibliografía

- [Abramowitz and Stegun, 1972] Abramowitz, M. and Stegun, I. A., editors (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, chapter Miscellaneous Functions, pages 997–1010. Dover Publications, New York, 9th edition.
- [Ahn et al., 2006] Ahn, C., Ramakrishna, R., and Goldberg, D. (2006). Real-coded bayesian optimization algorithm. In Lozano, J., Larrañaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 51–73. Springer Berlin / Heidelberg.
- [Baluja, 1994] Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University.
- [Bonet et al., 1997] Bonet, J. S. D., Isbell, C. L., and Viola, P. (1997). Mimic: Finding optima by estimating probability densities. In *Advances in Neural Information Processing Systems*, page 424. The MIT Press.
- [Bosman and Thierens, 2000a] Bosman, P. and Thierens, D. (2000a). Mixed ideas. Technical Report UU-CS-2000-45, Department of Information and Computing Sciences, Utrecht University.
- [Bosman, 2003] Bosman, P. A. (2003). *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*. PhD thesis, Universiteit Utrecht, Utrecht, The Netherlands.
- [Bosman and Thierens, 2000b] Bosman, P. A. N. and Thierens, D. (2000b). Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 197–200.

- [Chen et al., 2006] Chen, C.-H., Liu, W.-N., and Chen, Y.-P. (2006). Adaptive discretization for probabilistic model building genetic algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1103–1110, New York, NY, USA. ACM.
- [Cherubini et al., 2004] Cherubini, U., Luciano, E., and Vecchiato, W. (2004). *Copula methods in finance*. Wiley finance series. Wiley, Chichester [u.a.].
- [Cran et al., 1977] Cran, G. W., Martin, K. J., and Thomas, G. E. (1977). Remark as r19 and algorithm as 109: A remark on algorithms: As 63: The incomplete beta integral as 64: Inverse of the incomplete beta function ratio. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 26(1):pp. 111–114.
- [Cuesta-Infante et al., 2010] Cuesta-Infante, A., Santana, R., Hidalgo, J., Bielza, C., and Larrañandaga, P. (2010). Bivariate empirical and n-variate archimedean copulas in estimation of distribution algorithms. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8.
- [Ding et al., 2006] Ding, N., Zhou, S., and Sun, Z. (2006). Optimizing continuous problems using estimation of distribution algorithm based on histogram model. In Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H., Iba, H., Chen, G.-L., and Yao, X., editors, *Simulated Evolution and Learning*, volume 4247 of *Lecture Notes in Computer Science*, pages 545–552. Springer Berlin / Heidelberg.
- [Ding et al., 2008] Ding, N., Zhou, S.-D., and Sun, Z.-Q. (2008). Histogram-based estimation of distribution algorithm: A competent method for continuous optimization. *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, 23(1):35–43.
- [Durante and Sempi, 2010] Durante, F. and Sempi, C. (2010). Copula theory: An introduction. In Jaworski, P., Durante, F., Härdle, W. K., and Rychlik, T., editors, *Copula Theory and Its Applications*, volume 198 of *Lecture Notes in Statistics*, pages 3–31. Springer Berlin Heidelberg.

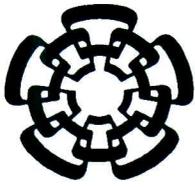
- [Elidan, 2010] Elidan, G. (2010). Copula bayesian networks. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 559–567. Curran Associates Inc.
- [Gallagher, 2000] Gallagher, M. R. (2000). *Multi-Layer Perceptron Error Surfaces: Visualization, Structure and Modelling Models for Iterative Global Optimization*. PhD thesis, University of Queensland, Queensland, Australia.
- [Genest and MacKay, 1986] Genest, C. and MacKay, J. (1986). The joy of copulas: Bivariate distributions with uniform marginals. *The American Statistician*, 40(4):pp. 280–283.
- [Guo et al., 2011] Guo, X., Wang, L., Zeng, J., and Zhang, X. (2011). Copula estimation of distribution algorithm with pmle. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, volume 2, pages 1077 –1081.
- [Hansen and Ostermeier, 2001] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195.
- [Harik et al., 2006] Harik, G., Lobo, F., and Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ecga). In Pelikan, M., Sastry, K., and CantúPaz, E., editors, *Scalable Optimization via Probabilistic Modeling*, volume 33 of *Studies in Computational Intelligence*, pages 39–61. Springer Berlin / Heidelberg.
- [Harik and Lobo, 1999] Harik, G. R. and Lobo, F. G. (1999). A parameter-less genetic algorithm. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 258–265, Orlando, Florida, USA. Morgan Kaufmann.
- [Joe and Xu, 1996] Joe, H. and Xu, J. (1996). The estimation method of inference functions for margins for multivariate models. Technical Report 166, Department of Statistics, University of British Columbia.

- [Johnson et al., 1995] Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995). *Continuous Univariate Distributions*, volume 2 of *Probability and Statistics*. Wiley-Interscience, 2 edition.
- [Kendall, 1938] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- [Kurowicka and Cooke, 2002] Kurowicka, D. and Cooke, R. M. (2002). Advanced input modeling: the vine copula method for representing high dimensional dependent distributions: application to continuous belief nets. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, WSC '02, pages 270–278. Winter Simulation Conference.
- [Larrañaga et al., 1999] Larrañaga, P., Etxeberria, R., Lozano, J. A., Peña, J., and Peña, J. M. (1999). Optimization by learning and simulation of bayesian and gaussian networks. Technical Report EHU-KZAA-IK-4/99, University of the Basque Country.
- [Larrañaga et al., 2000] Larrañaga, P., Etxeberria, R., Lozano, J. A., and Peña, J. M. (2000). Optimization in continuous domains by learning and simulation of gaussian networks. Technical report, University of the Basque Country.
- [Lobo and Lima, 2010] Lobo, F. G. and Lima, C. F. (2010). Towards automated selection of estimation of distribution algorithms. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, GECCO '10, pages 1945–1952, New York, NY, USA. ACM.
- [Nelsen, 1998] Nelsen, R. B. (1998). *An Introduction to Copulas (Lecture Notes in Statistics)*. Springer, 1 edition.
- [Ocenasek et al., 2004] Ocenasek, J., Kern, S., Hansen, N., and Koumoutsakos, P. (2004). A mixed bayesian optimization algorithm with variance adaptation. In Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tino, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 352–361. Springer Berlin / Heidelberg.

- [Ocenasek and Schwarz, 2002] Ocenasek, J. and Schwarz, J. (2002). Estimation of distribution algorithm for mixed continuous-discrete optimization problems. In *2nd Euro-International Symposium on Computational Intelligence*, pages 227–232, Košice, Slovakia.
- [Omidvar et al., 2010] Omidvar, M. N., Li, X., Yang, Z., and Yao, X. (2010). Cooperative co-evolution for large scale optimization through more frequent random grouping. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC2010)*, pages 1–8.
- [Pelikan et al., 1999] Pelikan, M., Goldberg, D. E., and Cantu-Paz, E. (1999). Boa: The bayesian optimization algorithm. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL. Morgan Kaufmann Publishers, San Francisco, CA.
- [Posten, 1993] Posten, H. O. (1993). An effective algorithm for the noncentral beta distribution function. *The American Statistician*, 47(2):pp. 129–131.
- [Rudlof and Köppen, 1997] Rudlof, S. and Köppen, M. (1997). Stochastic hill climbing with learning by vectors of normal distributions. In *Proceedings of the First Online Workshop on Soft Computinf (WSC1)*, pages 60–70.
- [Salinas-Gutiérrez et al., 2010] Salinas-Gutiérrez, R., Hernández-Aguirre, A., and Villa-Diharce, E. R. (2010). D-vine eda: a new estimation of distribution algorithm based on regular vines. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation, GECCO '10*, pages 359–366, New York, NY, USA. ACM.
- [Salinas-Gutiérrez et al., 2011] Salinas-Gutiérrez, R., Hernández-Aguirre, A., and Villa-Diharce, E. R. (2011). Dependence trees with copula selection for continuous estimation of distribution algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO '11*, pages 585–592, New York, NY, USA. ACM.

- [Santana et al., 2008] Santana, R., Larrañaga, P., and Lozano, J. (2008). Adaptive estimation of distribution algorithms. In Cotta, C., Sevaux, M., and Sörensen, K., editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pages 177–197. Springer Berlin / Heidelberg.
- [Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- [Sebag and Ducoulombier, 1998] Sebag, M. and Ducoulombier, A. (1998). Extending population-based incremental learning to continuous search spaces. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, PPSN V*, pages 418–427, London, UK. Springer-Verlag.
- [Servet et al., 1998] Servet, I., Travé-Massuyès, L., and Stern, D. (1998). Telephone network traffic overloading diagnosis and evolutionary computation techniques. In Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., and Snyers, D., editors, *Artificial Evolution*, volume 1363 of *Lecture Notes in Computer Science*, pages 137–144. Springer Berlin / Heidelberg. 10.1007/BFb0026596.
- [Sklar, 1959] Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de statistique de l'Université de Paris*, 8:229–231.
- [Strelen and Nassaj, 2007] Strelen, J. C. and Nassaj, F. (2007). Analysis and generation of random vectors with copulas. In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come, WSC '07*, pages 488–496, Piscataway, NJ, USA. IEEE Press.
- [Tang et al., 2010] Tang, K., Li, X., Suganthan, P. N., Yang, Z., , and Weise, T. (2010). Benchmark functions for the cec 2010 special session and competition on large-scale global optimization. Technical report.
- [Tsutsui et al., 2001] Tsutsui, S., Pelikan, M., and Goldberg, D. E. (2001). Evolutionary algorithm using marginal histogram models in continuous domain. In *Continuous Domain, Proc. of the 2001 Genetic and Evolutionary Computation Conference Workshop Program*, pages 230–233.

- [Wang et al., 2010a] Wang, L., Guo, X., Zeng, J., and Hong, Y. (2010a). Using gumbel copula and empirical marginal distribution in estimation of distribution algorithm. In *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on*, pages 583–587.
- [Wang et al., 2010b] Wang, L. F., Wang, Y. C., Zeng, J. C., and Hong, Y. (2010b). An estimation of distribution algorithm based on clayton copula and empirical margins. In Li, K., Li, X., Ma, S., and Irwin, G. W., editors, *Life System Modeling and Intelligent Computing*, volume 98 of *Communications in Computer and Information Science*, pages 82–88. Springer Berlin Heidelberg.
- [Wang and Zeng, 2010] Wang, L.-F. and Zeng, J.-C. (2010). Estimation of distribution algorithm based on copula theory. In Chen, Y.-p., Hiot, L. M., and Ong, Y. S., editors, *Exploitation of Linkage Learning in Evolutionary Algorithms*, volume 3 of *Adaptation, Learning, and Optimization*, pages 139–162. Springer Berlin Heidelberg.
- [Wang et al., 2009a] Wang, L.-F., Zeng, J.-C., and Hong, Y. (2009a). Estimation of distribution algorithm based on archimedean copulas. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC '09*, pages 993–996, New York, NY, USA. ACM.
- [Wang et al., 2009b] Wang, L.-F., Zeng, J.-C., and Hong, Y. (2009b). Estimation of distribution algorithm based on copula theory. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 1057–1063.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL IPN

UNIDAD TAMAULIPAS

Cd. Victoria, Tamaulipas, a 14 de diciembre de 2012.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará el C. MARIO FLORES MENDEZ, declaramos que hemos revisado la tesis titulada:

“Optimización Numérica Mediante Algoritmos de Estimación de Distribución”

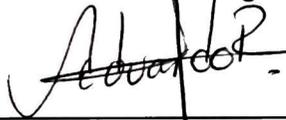
Y consideramos que cumple con los requisitos para obtener el grado de Maestro en Ciencias en Computación.

Atentamente,

Dr. Gregorio Toscano Pulido



Dr. Eduardo Arturo Rodríguez Tello



Dr. Ricardo Landa Becerra





CINVESTAV - IPN
Biblioteca Central



SSIT0011354