



Entrega Final

Análisis y Diseño de Algoritmos

Mgtr. Luz Enith Guerrero Mendieta



Índice general

I	Fundamentos Teóricos y Contextualización	
1	Representación Geométrica	9
1.1	Formalización del Problema de Bipartición Óptima	9
1.1.1	Limitaciones de los Enfoques de con Biparticiones Tradicionales	9
1.1.2	Correspondencia entre Sistemas Discretos y Espacios Geométricos	10
2	Tensores y Espacios N-dimensionales	13
2.1	Descomposición del Sistema en Tensores Elementales	13
2.1.1	Tensor de Probabilidad Condicional	14
2.1.2	Estructura Geométrica de los Tensores Elementales	14
2.2	Propiedades Topológicas del Espacio de Estados	15
2.2.1	Distancia Hamming y Estructura de Adyacencia	15
2.2.2	Invariancia Dimensional y Transformaciones Geométricas	16
II	Metodología Propuesta	
3	Reformulación Geométrica	23
3.1	Función de Costo para Transiciones entre Estados	23
3.1.1	Factores de Decrecimiento Exponencial	24
3.1.2	Exploración Recursiva del Espacio de Estados	25
3.2	Evaluación de Biparticiones mediante Discrepancia Tensorial	26
3.2.1	Distribuciones Marginales y Proyecciones Geométricas	26

4	Aplicación Práctica: Caso de Estudio	29
4.1	Análisis de un subsistema de Tres Variables	29
4.1.1	Construcción del Espacio Geométrico	29
4.1.2	Descomposición en Tensores Elementales	30
4.2	Cálculo de Función de Costo y Exploración del Espacio	31
4.2.1	Ejemplo de Transición: Estado 000 a Estado 011	31
4.2.2	Construcción Sistemática de la Tabla de Costos	31
4.2.3	Resultados Completos de la Tabla de Costos	35
4.2.4	Identificación de Biparticiones Óptimas	36
5	Entregables y Expectativas	37
5.1	Componentes Esenciales de la Implementación	37
5.1.1	Módulos de Software Requeridos	37
5.1.2	Interfaces y Estructuras de Datos	39
5.2	Criterios de Evaluación y Casos de Prueba	39
5.2.1	Conjunto de Datos de Prueba	39
5.2.2	Métricas de Desempeño	40
6	Perspectivas y Desarrollos Futuros	43
6.1	Extensiones Teóricas del Enfoque Geométrico	43
6.2	Innovaciones Algorítmicas y Computacionales	44
6.2.1	Optimización Heurística Basada en Propiedades Topológicas	44
6.2.2	Implementación Eficiente y Escalable	44



Fundamentos Teóricos y Contextualización

1	Representación Geométrica	9
1.1	Formalización del Problema de Bipartición Óptima	
2	Tensores y Espacios N-dimensionales	13
2.1	Descomposición del Sistema en Tensores Elementales	
2.2	Propiedades Topológicas del Espacio de Estados	

En el análisis de **sistemas complejos**, la identificación de componentes funcionales y sus interacciones representa un desafío fundamental. Los sistemas deterministas caracterizados por variables binarias que evolucionan en el tiempo constituyen un marco matemático ampliamente utilizado para **modelar** estos fenómenos. Este trabajo introduce un enfoque **geométrico-topológico** para abordar el problema de bipartición óptima, estableciendo nuevas metodologías que reducen la complejidad computacional inherente a los algoritmos tradicionales.



1. Representación Geométrica

Los sistemas deterministas compuestos por variables binarias han sido tradicionalmente analizados mediante matrices de probabilidad de transición y técnicas combinatorias. Este capítulo introduce una perspectiva geométrica alternativa, donde los estados del sistema se representan como vértices en un espacio n -dimensional, permitiendo aplicar herramientas de la topología algebraica para reducir la complejidad computacional del problema.

La fundamentación de este enfoque se basa en establecer una correspondencia biyectiva entre el espacio de estados binarios y la estructura geométrica de un hipercubo unitario. Esta representación no solo facilita la visualización del sistema, sino que también revela propiedades estructurales que pueden ser explotadas algorítmicamente para la identificación eficiente de biparticiones óptimas.

1.1 Formalización del Problema de Bipartición Óptima

Consideremos un sistema $V_o = \{X_1, X_2, \dots, X_n\}$ compuesto por n variables binarias. La evolución temporal de este sistema se representa mediante una Matriz de Probabilidad de Transición (TPM) que codifica $P(V_{t+1}|V_t)$, es decir, la probabilidad de transición entre estados en tiempos consecutivos.

El problema de bipartición óptima consiste en encontrar una división del sistema $V_o = S_1 \cup S_2$, con $S_1 \cap S_2 = \emptyset$, tal que se minimice la discrepancia entre la dinámica del sistema original y la dinámica reconstruida a partir de las partes:

$$\delta(V, \{S_1, S_2\}) = g(P(V_{t+1}|V_t), P(S_{1,t+1}|S_{1,t}) \otimes P(S_{2,t+1}|S_{2,t})) \quad (1.1)$$

donde \otimes denota el producto tensorial y d es una métrica adecuada en el espacio de distribuciones de probabilidad, típicamente basada en la distancia Earth Mover's Distance (EMD) con métrica de Hamming como costo de transporte.

1.1.1 Limitaciones de los Enfoques de con Biparticiones Tradicionales

Los enfoques tradicionales para resolver el problema de bipartición óptima se basan principalmente en estrategias combinatorias que evalúan explícitamente todas las posibles biparticiones o

utilizan heurísticas para explorar un subconjunto del espacio de soluciones. Sin embargo, estos métodos enfrentan limitaciones significativas cuando aumenta la dimensión del sistema.

Estrategias Actuales y sus Limitaciones Computacionales

Actualmente, las metodologías predominantes para abordar el problema de bipartición incluyen:

- **Búsqueda exhaustiva:** Implementada en bibliotecas como PyPhi, evalúa todas las posibles biparticiones del sistema (*Fuerza Bruta*). Aunque garantiza encontrar la solución óptima, su complejidad la hace inviable para sistemas con más de 20-25 variables.
- **Algoritmos heurísticos:** Como QNodes, emplean estrategias de selección para explorar un subconjunto del espacio de soluciones. Si bien ofrece ciertas garantías sobre la optimalidad del resultado, se puede reducir en tiempo de cómputo.
- **Técnicas de optimización:** Utilizan métodos como algoritmos genéticos o simulated annealing para aproximar la solución óptima, pero requieren múltiples evaluaciones de la función objetivo, lo que sigue siendo computacionalmente costoso.

El proceso tradicional inicia con el sistema original V_o , aplica condiciones de fondo para generar un sistema candidato V_c , marginalizaciones para obtener un subsistema V_s y posteriormente genera y evalúa biparticiones sobre este subsistema. El cuello de botella se encuentra precisamente en esta última etapa, donde la generación y evaluación de biparticiones domina la complejidad computacional del proceso de forma tangible e intangible.

Crecimiento Exponencial del Espacio de Búsqueda

La principal limitación de los enfoques tradicionales radica en la explosión combinatoria del espacio de búsqueda. Para un subsistema con u elementos en tiempo t y v elementos en tiempo $t + 1$, el número de posibles biparticiones se expresa como:

$$\mathbb{P}_2(u, v) = 2^{u+v-1} - 1 \quad (1.2)$$

Esta expresión revela un crecimiento exponencial con respecto al número total de variables del sistema. Para ilustrar esta explosión combinatoria, consideremos:

Tamaño del sistema ($u + v$)	Número de biparticiones	Tiempo estimado
4	7	milisegundos
8	127	segundos
16	32,767	horas
32	~2.1 billones	años

Cuadro 1.1: Crecimiento exponencial del espacio de búsqueda y su impacto en el tiempo de computación.

Este crecimiento exponencial hace que los enfoques de fuerza bruta sean impracticables para sistemas de tamaño moderado a grande y las heurísticas existentes no logran reducir suficientemente la complejidad para sistemas realmente complejos.

La metodología que proponemos busca analizar el subsistema **antes** de generar cualquier bipartición, aprovechando propiedades geométricas para identificar directamente biparticiones candidatas de alta calidad, o en el mejor caso, la bipartición óptima sin necesidad de evaluar exhaustivamente el espacio de soluciones.

1.1.2 Correspondencia entre Sistemas Discretos y Espacios Geométricos

El enfoque propuesto se fundamenta en una correspondencia natural entre sistemas de variables binarias y espacios geométricos n -dimensionales. Esta correspondencia permite reinterpretar el

problema de bipartición en términos topológicos, aprovechando propiedades geométricas que no son evidentes en la formulación tradicional.

Mapeo entre Estados Binarios y Coordenadas Espaciales

Existe una correspondencia biyectiva natural entre un sistema de n variables binarias y los vértices de un hipercubo n -dimensional. Formalmente, definimos una función de mapeo $\beta : \{0, 1\}^n \rightarrow \mathbb{R}^n$ tal que:

$$\beta(X_1, X_2, \dots, X_n) = (X_1, X_2, \dots, X_n) \quad (1.3)$$

Bajo este mapeo:

- Cada variable X_i corresponde a una dimensión en el espacio euclidiano \mathbb{R}^n .
- Cada estado posible del sistema (x_1, x_2, \dots, x_n) donde $x_i \in \{0, 1\}$ corresponde a un vértice del hipercubo unitario.
- Los 2^n posibles estados del sistema se mapean exactamente a los 2^n vértices del hipercubo n -dimensional.
- Dos estados son adyacentes en el sentido de Hamming (difieren en exactamente una variable) si y solo si sus correspondientes vértices están conectados por una arista en el hipercubo.

Este mapeo preserva la estructura combinatoria del sistema original, pero revela una interpretación geométrica que permite aplicar herramientas de la topología y la geometría computacional.

Propiedades Emergentes de la Representación Espacial

La representación geométrica del sistema revela propiedades estructurales que no son inmediatamente evidentes en la formulación matricial tradicional:

- **Distancia métrica:** La distancia de Hamming entre estados se corresponde con la distancia Manhattan entre vértices del hipercubo, proporcionando una medida natural de cercanía.^{entre} estados.
- **Estructura de vecindad:** La estructura de adyacencia del hipercubo codifica naturalmente las transiciones que involucran el cambio de una sola variable, formando la base para un análisis de transiciones '*elementales*'.
- **Simetría y regularidad:** El hipercubo posee propiedades de simetría que pueden explotarse para reducir el espacio de búsqueda, identificando clases de equivalencia entre biparticiones.
- **Descomposición dimensional:** El hipercubo n -dimensional puede proporcionar una *interpretación geométrica natural* para las biparticiones del sistema por simetría.

Estas propiedades permiten reformular el problema de bipartición como un problema de complementación del hipercubo, donde buscamos la estructura que minimice cierta función de costo topológica relacionada con la discrepancia δ definida anteriormente.

La ventaja fundamental de esta reformulación es que podemos explotar propiedades geométricas y topológicas para identificar candidatos a biparticiones óptimas sin necesidad de evaluar explícitamente todas las posibles particiones, reduciendo drásticamente la complejidad computacional del problema.



2. Tensores y Espacios N-dimensionales

Tras establecer la correspondencia fundamental entre sistemas discretos y espacios geométricos, profundizamos ahora en la estructura algebraica subyacente mediante la representación tensorial. Esta perspectiva permite descomponer la dinámica del sistema en componentes elementales que preservan las propiedades probabilísticas, facilitando el análisis de interacciones causales y la evaluación eficiente de biparticiones.

La representación tensorial no solo ofrece una formalización matemática rigurosa, sino que también establece conexiones con estructuras de datos multidimensionales utilizadas en análisis computacional avanzado. Este enfoque unifica conceptos de álgebra multilineal, teoría de la probabilidad y geometría computacional, proporcionando un marco teórico sólido para el análisis de sistemas complejos.

2.1 Descomposición del Sistema en Tensores Elementales

Un aspecto fundamental de nuestro enfoque es la descomposición del sistema en tensores elementales, cada uno asociado a una variable específica en tiempo $t + 1$. Esta descomposición, basada en la independencia condicional entre variables en tiempo futuro dado el estado completo en tiempo presente, permite analizar la contribución individual de cada componente a la dinámica global.

Para un sistema de n variables, la matriz de probabilidad de transición (TPM) puede representarse de dos formas principales:

- **Forma estado-estado:** Donde las filas representan los 2^n posibles estados en tiempo t y las columnas los 2^n estados en tiempo $t + 1$.
- **Forma estado-nodo:** Donde las filas siguen representando los estados en tiempo t , pero las columnas se reorganizan para representar los valores (0 ó 1) de cada variable individual en tiempo $t + 1$.

La forma estado-nodo resulta particularmente útil para la descomposición tensorial, ya que separa la TPM en n matrices más pequeñas, cada una codificando la probabilidad condicional de una variable específica en tiempo futuro.

2.1.1 Tensor de Probabilidad Condicional

El tensor de probabilidad condicional es una estructura matemática que generaliza la noción de matriz para capturar las relaciones probabilísticas multivariadas en el sistema. Para cada variable X_i en tiempo $t + 1$, definimos un tensor que codifica $P(X_i^{t+1}|V^t)$.

Relación con Matrices de Transición

Los tensores elementales están intrínsecamente relacionados con la matriz de probabilidad de transición (TPM) tradicional mediante el principio de independencia condicional. Este principio establece que las probabilidades de transición para diferentes variables en tiempo $t + 1$ son independientes entre sí, condicionadas al estado completo del sistema en tiempo t .

Formalmente, para un sistema con n variables, la relación entre la TPM completa y los tensores elementales se expresa mediante el producto tensorial:

$$P(V^{t+1}|V^t) = P(X_1^{t+1}|V^t) \otimes P(X_2^{t+1}|V^t) \otimes \dots \otimes P(X_n^{t+1}|V^t) \quad (2.1)$$

donde \otimes denota el producto tensorial entre las distribuciones de probabilidad condicional.

Esta descomposición permite reconstruir la TPM completa (de tamaño $2^n \times 2^n$) a partir de n matrices más pequeñas (cada una de tamaño $2^n \times 2$), lo que representa una ventaja tanto computacional como conceptual para el análisis del sistema.

2.1.2 Estructura Geométrica de los Tensores Elementales

Cada tensor elemental puede interpretarse geométricamente como una función definida sobre los vértices de un hipercubo n -dimensional. Esta interpretación establece un puente conceptual entre la representación algebraica (tensores) y la representación geométrica (hipercubo) del sistema.

Analogía con Cubos OLAP

La estructura tensorial que empleamos guarda una estrecha relación con los cubos OLAP (OnLine Analytical Processing) utilizados en ciencia de datos y análisis multidimensional. En este contexto:

- Las dimensiones del tensor corresponden a las dimensiones del cubo OLAP.
- Los índices específicos corresponden a coordenadas dentro del cubo.
- Los valores almacenados representan medidas o métricas (en nuestro caso, probabilidades condicionales).

Esta analogía permite aplicar operaciones fundamentales de OLAP al análisis de sistemas causales:

- **Slice:** Extracción de un subconjunto del tensor fijando el valor de una o más dimensiones, correspondiente a condicionar el sistema.
- **Dice:** Selección de un subcubo mediante restricciones en múltiples dimensiones, similar a la marginalización en un subconjunto de variables.
- **Roll-up:** Agregación de datos a lo largo de una jerarquía dimensional, análogo a la marginalización sobre variables específicas.
- **Drill-down:** Descomposición de datos agregados en componentes más detallados, comparable a la expansión de variables compuestas.

¡Esta correspondencia enriquece nuestro arsenal **metodológico y tecnológico**! Permitiendo adaptar técnicas y algoritmos del campo de análisis de datos multidimensionales al problema específico de bipartición óptima (*Se recomienda ver una Introducción a los Cubos OLAP para mayor comprensión*).

Visualización Progresiva de Dimensiones

La interpretación geométrica de los tensores elementales se puede visualizar mediante la progresión de estructuras n-dimensionales:

- **0-cubo (punto):** Corresponde a un sistema sin variables, donde el "tensor" es simplemente un escalar.
- **1-cubo (línea):** Representa un sistema con una variable binaria, donde el tensor se visualiza como un vector de dos componentes (probabilidades para los estados 0 y 1).
- **2-cubo (cuadrado):** Modela un sistema con dos variables, donde el tensor se visualiza como una matriz 2×2 .
- **3-cubo (cubo):** Corresponde a un sistema con tres variables, donde el tensor puede representarse como un arreglo tridimensional $2 \times 2 \times 2$.
- **n-cubo (hipercubo):** Generaliza la estructura para sistemas con n variables.

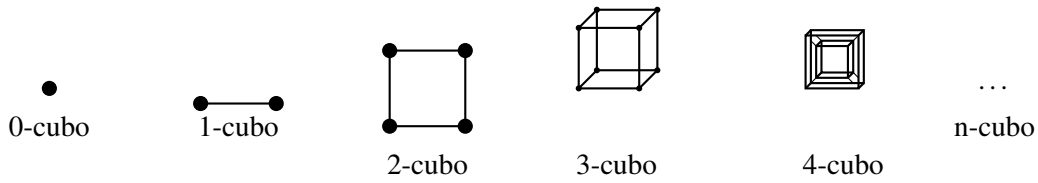


Figura 2.1: Progresión de estructuras dimensionales desde el 0-cubo hasta el n-cubo, ilustrando la generalización geométrica que subyace a la representación tensorial de sistemas con múltiples variables.

Esta progresión dimensional intenta proporcionar una visualización intuitiva que también ayude a comprender cómo escalan las propiedades y operaciones al aumentar la dimensionalidad del sistema. Las propiedades que son evidentes en dimensiones bajas (como la conectividad en un grafo) pueden generalizarse a dimensiones superiores, guiando el desarrollo de algoritmos eficientes para sistemas complejos.

2.2 Propiedades Topológicas del Espacio de Estados

La representación geométrica del sistema como hipercono induce una estructura topológica rica que determina las propiedades esenciales del espacio de estados. Estas propiedades topológicas no son meros accesorios matemáticos; constituyen la base para desarrollar algoritmos eficientes para la identificación de biparticiones óptimas.

2.2.1 Distancia Hamming y Estructura de Adyacencia

La distancia de Hamming entre dos estados binarios se define como el número de posiciones en las que difieren sus representaciones binarias. En el contexto del hipercono, esta distancia corresponde exactamente al número mínimo de aristas que deben recorrerse para ir de un vértice a otro.

Cálculo e Interpretación de la Distancia Hamming

Formalmente, para dos estados $x = (x_1, x_2, \dots, x_n)$ y $y = (y_1, y_2, \dots, y_n)$ donde $x_i, y_i \in \{0, 1\}$, la distancia de Hamming se define como:

$$d_H(x, y) = \sum_{i=1}^n |x_i - y_i| = \sum_{i=1}^n x_i \oplus y_i \quad (2.2)$$

donde \oplus denota la operación XOR (disyunción exclusiva).

Esta distancia tiene una interpretación natural en términos de transiciones entre estados: representa el número mínimo de variables que deben cambiar su valor para transformar un estado en otro. Por ejemplo:

- $d_H(000, 001) = 1$ - Solo difieren en la última posición
- $d_H(010, 101) = 2$ - Difieren en la primera y segunda posición
- $d_H(000, 111) = 3$ - Difieren en todas las posiciones

La estructura de adyacencia del hipercubo está completamente determinada por la distancia de Hamming: dos vértices son adyacentes (conectados por una arista) si y solo si su distancia de Hamming es exactamente 1.

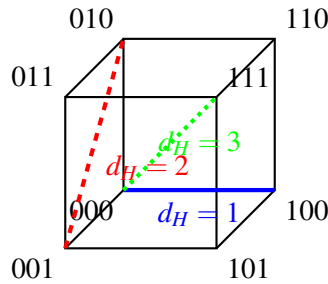


Figura 2.2: Representación de un 3-cubo con vértices etiquetados mediante coordenadas binarias. Se ilustran distancias de Hamming $d_H = 1$ (azul, sólido), $d_H = 2$ (rojo, discontinuo) y $d_H = 3$ (verde, punteado) desde el vértice 000.

Rutas Óptimas en el Espacio de Estados

La distancia de Hamming induce una métrica natural en el espacio de estados, permitiendo definir y calcular rutas óptimas entre estados. Una ruta óptima entre dos estados corresponde a un camino de longitud mínima en el hipercubo, donde cada paso implica cambiar el valor de exactamente una variable.

2.2.2 Invariancia Dimensional y Transformaciones Geométricas

La estructura del hipercubo exhibe propiedades de invariancia bajo ciertas transformaciones, lo que permite identificar equivalencias entre configuraciones aparentemente distintas del sistema.

Transformaciones que Preservan la Estructura

Entre las transformaciones que preservan la estructura topológica del hipercubo se incluyen:

- **Permutaciones de coordenadas:** Corresponden a reordenar las variables del sistema sin alterar sus relaciones funcionales.
- **Complementación de coordenadas:** Equivale a invertir la interpretación de una variable (intercambiar los valores 0 y 1).
- **Automorfismos del hipercubo:** Transformaciones que preservan la estructura de adyacencia, generadas por combinaciones de las operaciones anteriores.

Estas transformaciones forman el grupo de simetría del hipercubo, cuyo orden es $2^n \cdot n!$ para un n-cubo, reflejando las 2^n posibles inversiones de coordenadas y las $n!$ permutaciones de las n dimensiones.

Implicaciones para el Análisis del Sistema

Las propiedades de invariancia dimensional tienen profundas implicaciones para el análisis de biparticiones:

- **Reducción del espacio de búsqueda:** Muchas biparticiones aparentemente distintas son equivalentes bajo transformaciones que preservan la estructura, permitiendo reducir significativamente el número de configuraciones a evaluar.
- **Identificación de patrones estructurales:** Las simetrías del hipercubo revelan patrones estructurales en la dinámica del sistema que pueden explotarse para identificar biparticiones óptimas.
- **Generalización de resultados:** Los resultados obtenidos para ciertos subsistemas pueden generalizarse a otros subsistemas equivalentes bajo transformaciones adecuadas.

Al explotar estas propiedades de invariancia, podemos desarrollar algoritmos más eficientes que exploren selectivamente el espacio de soluciones, enfocándose en representantes canónicos de clases de equivalencia en lugar de evaluar cada configuración individualmente.

La identificación y explotación de estas propiedades topológicas constituye uno de los pilares fundamentales de nuestro enfoque, permitiendo reducir la complejidad computacional inherente al problema de bipartición óptima sin sacrificar la calidad de la solución encontrada.



Metodología Propuesta

3	Reformulación Geométrica	23
3.1	Función de Costo para Transiciones entre Estados	
3.2	Evaluación de Biparticiones mediante Discrepancia Tensorial	
4	Aplicación Práctica: Caso de Estudio	29
4.1	Análisis de un subsistema de Tres Variables	
4.2	Cálculo de Función de Costo y Exploración del Espacio	
5	Entregables y Expectativas	37
5.1	Componentes Esenciales de la Implementación	
5.2	Criterios de Evaluación y Casos de Prueba	
6	Perspectivas y Desarrollos Futuros	43
6.1	Extensiones Teóricas del Enfoque Geométrico	
6.2	Innovaciones Algorítmicas y Computacionales	

En esta sección se da la propuesta metodológica basada en la representación geométrica. Partiendo de la correspondencia establecida entre el sistema discreto y el espacio n -dimensional, desarrollamos un enfoque que reformula el problema de bipartición óptima en términos de la estructura topológica del hipercubo y sus propiedades intrínsecas.

La metodología propuesta se fundamenta en dos pilares principales: **(1)** el análisis de transiciones entre estados mediante una función de costo topológicamente informada, y **(2)** la evaluación de biparticiones candidatas utilizando propiedades de invariancia y distribuciones marginales, evitando la necesidad de reconstruir explícitamente el sistema completo.

3. Reformulación Geométrica

Habiendo establecido la correspondencia entre el sistema discreto y el espacio geométrico n -dimensional, procedemos ahora a reformular el problema de bipartición óptima aprovechando esta interpretación espacial. La reformulación geométrica no solo proporciona una nueva perspectiva conceptual, sino que también establece las bases para un enfoque algorítmico más eficiente.

El problema tradicional de encontrar una bipartición óptima del sistema se transforma en un problema de análisis topológico del hipercubo asociado, donde buscamos identificar una división que preserve óptimamente ciertas propiedades estructurales. Esta transformación nos permite explotar características intrínsecas del espacio de estados que no son evidentes en la formulación original.

3.1 Función de Costo para Transiciones entre Estados

Un componente fundamental es la definición de una función de costo $t(i, j)$ que cuantifica la 'inercia' o 'energía' requerida para la **transición** entre dos estados i, j del sistema. Esta función captura la estructura causal subyacente al sistema y proporciona la base para identificar biparticiones naturales.

A diferencia de métricas tradicionales que consideren solo transiciones directas entre estados, $t(i, j)$ integra el concepto de distancia topológica en el hipercubo, reconociendo que hay transiciones entre estados distantes implican múltiples transformaciones elementales intermedias.

Formalmente se define la función de costo $t : V \times V \rightarrow \mathbb{R}^+$ para la transición entre estados i, j como:

$$t_x(i, j) = \gamma \cdot \left(|X[i] - X[j]| + \sum_{k \in N(i, j)} \{t(k, j)\} \right) \quad (3.1)$$

donde:

- $d(i, j)$ es la distancia de Hamming entre los estados i y j .
- $\gamma = 2^{-d(i, j)}$ es un factor de decrecimiento exponencial basado en una distancia métrica (Hamming).

- $X[i]$ representa el valor asociado al estado i (*probabilidad condicional*).
- $N(i)$ denota el conjunto de vecinos inmediatos de i en el hipercubo.

La tabla \mathbb{T} resultante de aplicar esta función a todos los pares de estados constituye un mapa completo de las relaciones causales en el sistema, revelando patrones estructurales que guiarán la identificación de biparticiones óptimas.

3.1.1 Factores de Decrecimiento Exponencial

Un aspecto crítico de la función de costo $t(i, j)$ es el factor de decrecimiento exponencial $\gamma = 2^{-d(i, j)}$, que determina cómo la influencia de las diferencias entre estados disminuye con la distancia topológica.

Justificación del Factor Exponencial

La elección de un factor de decrecimiento exponencial, en lugar de otras funciones como lineal o logarítmica, se basa en consideraciones tanto teóricas como prácticas:

- **Fundamentación teórica:** En sistemas causales, la influencia de un evento sobre otro típicamente decae exponencialmente con la "distancia" (temporal, espacial o topológica) entre ellos. Este comportamiento se observa en fenómenos físicos, biológicos y sociales.
- **Propiedades matemáticas:** La función exponencial 2^{-d} posee propiedades algebraicas convenientes, como el hecho de que $2^{-(d_1+d_2)} = 2^{-d_1} \cdot 2^{-d_2}$, lo que permite descomponer transiciones complejas en componentes elementales. Así mismo, presenta una correlación con el cálculo de la EMD arraigada en su naturaleza.
- **Comportamiento asintótico:** Con esta función, la influencia nunca llega exactamente a cero pero se vuelve arbitrariamente pequeña para distancias grandes, reflejando el principio de que todo puede influir en todo, pero con magnitudes prácticas muy diferentes.

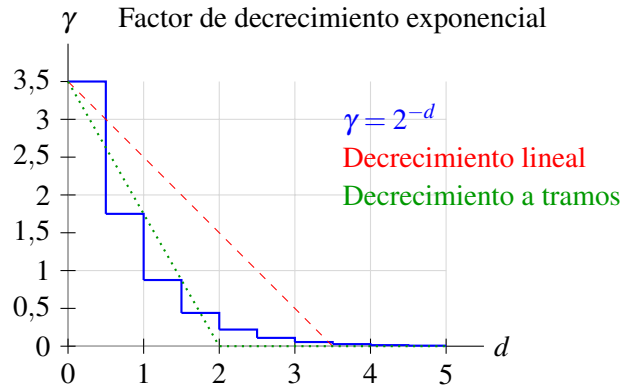


Figura 3.1: Comparación del factor de decrecimiento exponencial $\gamma = 2^{-d}$ con otros tipos de decrecimiento en función de la distancia d . El decrecimiento exponencial captura adecuadamente la dilución de la influencia causal con la distancia topológica.

Efectos sobre el Análisis de Transiciones

El factor exponencial tiene efectos significativos en el análisis de transiciones entre estados:

- **Ponderación natural de caminos:** Al calcular $t(i, j)$ mediante exploración recursiva del hipercubo, los caminos más cortos reciben automáticamente mayor peso en la contribución total.
- **Localización de efectos:** Las diferencias entre estados cercanos dominan el cálculo, reflejando la intuición de que las interacciones locales son generalmente más fuertes que las distantes.

Esta ponderación exponencial tiene un efecto particularmente importante en sistemas de alta dimensionalidad, donde el número de estados y posibles transiciones crece exponencialmente, pero solo un subconjunto de estas relaciones resulta significativo para la estructura causal del sistema.

3.1.2 Exploración Recursiva del Espacio de Estados

El cálculo de la función de costo $t(i, j)$ para todos los pares de estados requiere una exploración sistemática del hipercubo. Dado que el número de estados crece exponencialmente con la dimensionalidad del sistema, es crucial implementar un algoritmo eficiente para esta exploración.

Algoritmo de Búsqueda en Anchura Modificado

Se ha de plantear un algoritmo de Búsqueda en Anchura (BFS) modificado que explore el hipercubo nivel por nivel, acumulando contribuciones al costo con la ponderación adecuada:

Algorithm 1 Cálculo de la Tabla de Costos T mediante BFS Modificado

```

1: Input: Hipercubo  $G = (V, E)$ , valores  $X[v]$  para cada vértice  $v \in V$ 
2: Output: Tabla de costos  $T$  donde  $T[i, j] = t(i, j)$  para todo  $i, j \in V$ 
3: for cada par de vértices  $(i, j) \in V \times V$  do
4:    $T[i, j] \leftarrow 0$  ▷ Inicialización
5:    $d \leftarrow d_H(i, j)$  ▷ Distancia de Hamming
6:    $\gamma \leftarrow 2^{-d}$  ▷ Factor de decrecimiento
7:    $T[i, j] \leftarrow |X[i] - X[j]|$  ▷ Contribución directa
8:   if  $d > 1$  then ▷ Si no son vecinos inmediatos
9:      $Q \leftarrow \{i\}$  ▷ Cola para BFS
10:     $visited \leftarrow \{i\}$  ▷ Conjunto de vértices visitados
11:     $level \leftarrow 0$  ▷ Nivel actual en BFS
12:    while  $level < d$  and  $Q$  no vacío do
13:       $level \leftarrow level + 1$ 
14:       $nextQ \leftarrow \{\}$  ▷ Cola para el siguiente nivel
15:      for cada vértice  $u \in Q$  do
16:        for cada vecino  $v$  de  $u$  tal que  $d_H(v, j) < d_H(u, j)$  do
17:          if  $v \notin visited$  then
18:             $T[i, j] \leftarrow \gamma(T[i, j] + T[i, v])$  ▷ Acumular costo
19:             $visited \leftarrow visited \cup \{v\}$ 
20:             $nextQ \leftarrow nextQ \cup \{v\}$ 
21:          end if
22:        end for
23:      end for
24:       $Q \leftarrow nextQ$ 
25:    end while
26:  end if
27: end for
28: return  $T$ 

```

Este algoritmo construye la tabla completa de costos \mathbb{T} , capturando la estructura causal del sistema representada en el hipercubo. La tabla resultante sirve como base para la identificación y evaluación de biparticiones potenciales.

Optimizaciones para Sistemas Grandes

Para sistemas de gran escala, donde la dimensionalidad puede hacer prohibitivo el cálculo exhaustivo se pueden proponer optimizaciones como:

- **Paralelización:** El cálculo de $t(i, j)$ para diferentes pares (i, j) es inherentemente paralelizable, permitiendo una implementación eficiente en arquitecturas multicore o distribuidas.
- **Aproximación mediante muestreo:** Para sistemas extremadamente grandes, utilizar técnicas de muestreo para estimar los valores de $t(i, j)$ en lugar de calcularlos exactamente para todos los pares.
- **Aprovechar simetrías:** Utilizar las propiedades de simetría del hipercubo para reducir el número de cálculos necesarios, reconociendo que muchos pares de estados son equivalentes bajo transformaciones que preservan la estructura.

Estas optimizaciones permiten extender la aplicabilidad de nuestra metodología a sistemas con decenas o incluso cientos de variables, donde los enfoques tradicionales resultarían completamente inviables.

3.2 Evaluación de Biparticiones mediante Discrepancia Tensorial

Una vez calculada la tabla de costos \mathbb{T} , procedemos a utilizarla para evaluar la calidad de posibles biparticiones del sistema. A diferencia de los enfoques tradicionales que requieren reconstruir el sistema completo mediante productos tensoriales, nuestra metodología aprovecha propiedades geométricas y distribuciones marginales para una evaluación más directa y eficiente.

3.2.1 Distribuciones Marginales y Proyecciones Geométricas

Un concepto clave en nuestra metodología es el uso de distribuciones marginales como proyecciones geométricas en el espacio n -dimensional. Este enfoque elimina la necesidad de reconstruir explícitamente el sistema mediante productos tensoriales, reduciendo significativamente la complejidad computacional.

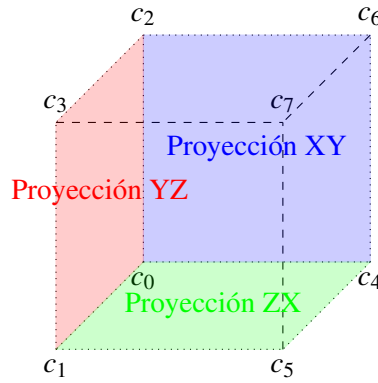


Figura 3.2: Visualización de distribuciones marginales como proyecciones geométricas en un sistema de tres variables. Las proyecciones en diferentes planos corresponden a marginalizar sobre distintos subconjuntos de variables.

Cuando consideramos una bipartición $S = \{S_1, S_2\}$ del sistema, cada parte S_i define un subespacio en el hipercubo n -dimensional. Las propiedades de estos subespacios y sus interrelaciones determinan la calidad de la bipartición.

La metodología tradicional evaluaría la discrepancia entre la dinámica del sistema original y la dinámica reconstruida mediante el producto tensorial de las partes. En contraste, nuestro enfoque explota el hecho de que esta discrepancia puede caracterizarse directamente mediante propiedades geométricas del hipercubo y las distribuciones marginales asociadas.

- **Proyección dimensional:** Una distribución marginal corresponde geoméricamente a una proyección del hipercubo sobre un subespacio de menor dimensión.

- **Conservación de información:** La proyección preserva cierta información sobre la estructura original, pero inevitablemente pierde algunas relaciones al reducir la dimensionalidad.
- **Independencia estructural:** Dos subconjuntos de variables son funcionalmente independientes si sus proyecciones correspondientes capturan toda la información relevante del sistema original.

La independencia funcional entre subconjuntos de variables se manifiesta geoméricamente como una propiedad de descomposición del hipercubo, donde las proyecciones sobre subespacios complementarios son suficientes para caracterizar completamente el sistema.



4. Aplicación Práctica: Caso de Estudio

Tras desarrollar el marco teórico y metodológico para la representación geométrica del problema de bipartición óptima, procedemos ahora a aplicar estos conceptos a un caso concreto. Este capítulo presenta un análisis detallado de un subsistema simple de tres variables, ilustrando paso a paso cómo nuestra metodología permite identificar biparticiones óptimas.

4.1 Análisis de un subsistema de Tres Variables

Consideremos un subsistema compuesto por tres variables binarias $V=\{A, B, C\}$. Este subsistema nos permitirá ilustrar los principios y permitir una visualización clara de los conceptos subyacentes.

4.1.1 Construcción del Espacio Geométrico

Modelado del subsistema Original

Nuestro subsistema de ejemplo consta de tres variables binarias (*pueden tomar los valores 0 o 1*). La dinámica del subsistema está definida por una matriz de probabilidad de transición (TPM) que especifica la probabilidad de cada posible estado en tiempo $t + 1$ dado cada posible estado en tiempo t (*Este conjunto de datos se puede encontrar en el proyecto como la Red de 03 Nodos C, "N3C.csv"*).

Para un subsistema de tres variables, la TPM completa en formato estado-estado sería una matriz 8×8 , donde las filas representan los $2^3 = 8$ posibles estados en tiempo t y las columnas representan los estados en tiempo $t + 1$. Sin embargo, para nuestro análisis basado en la representación geométrica, trabajaremos con la forma descompuesta estado-nodo, donde cada variable futura se representa por separado (*contendría dimensiones de 8×3 como se tabuló previamente*).

Representación como Hipercubo

La representación geométrica del subsistema corresponde a un hipercubo tridimensional (cubo), donde cada vértice representa uno de los $2^{|V|}=3 = 8$ posibles estados del subsistema, etiquetados según su representación binaria con el dataset en su estado OFF.

En esta representación:

		A_{t+1} (OFF)	B_{t+1} (OFF)	C_{t+1} (OFF)
abc_t	000	0	0	0
	100	0	0	1
	010	0	1	0
	110	0	1	1
	001	1	0	0
	101	1	0	1
	011	1	1	0
	111	1	1	1

Cuadro 4.1: TPM Representación estado-nodo

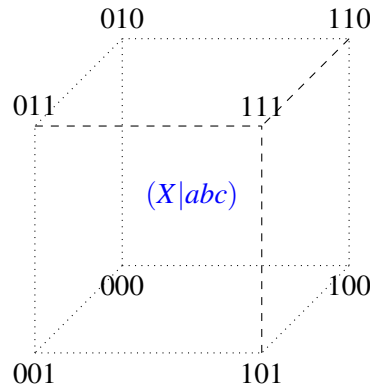


Figura 4.1: Representación del subsistema de tres variables como un cubo tridimensional. Cada vértice corresponde a un posible estado del subsistema, etiquetado según su representación binaria.

- El primer dígito corresponde al valor de la variable/dimensión a .
- El segundo dígito corresponde al valor de la variable/dimensión b .
- El tercer dígito corresponde al valor de la variable/dimensión c .

La estructura geométrica del cubo captura naturalmente la relación de adyacencia entre estados: dos estados son adyacentes (conectados por una arista) si y solo si difieren en exactamente una variable. Esta propiedad será fundamental para nuestro análisis de transiciones entre estados.

4.1.2 Descomposición en Tensores Elementales

Siguiendo el principio de independencia condicional, podemos descomponer la dinámica del subsistema en tres tensores elementales, cada uno representando la probabilidad condicional de una variable específica en tiempo $t + 1$ dado el estado completo del subsistema en tiempo t .

Tensores Individuales para Cada Variable

Para nuestro subsistema de ejemplo, consideremos los siguientes valores para los tensores elementales, que representan $P(A_{t+1}|abc_t)$, $P(B_{t+1}|abc_t)$ y $P(C_{t+1}|abc_t)$ respectivamente:

Esta estructura ilustra un subsistema con dependencias causales específicas entre variables en diferentes tiempos.

Verificación de la Descomposición

La descomposición tensorial nos permite reconstruir la dinámica completa del subsistema mediante el producto tensorial de los tensores elementales. Para cada estado inicial abc_t , podemos determinar la distribución de probabilidad sobre los estados ABC_{t+1} combinando las probabilidades condicionales de cada variable individual.

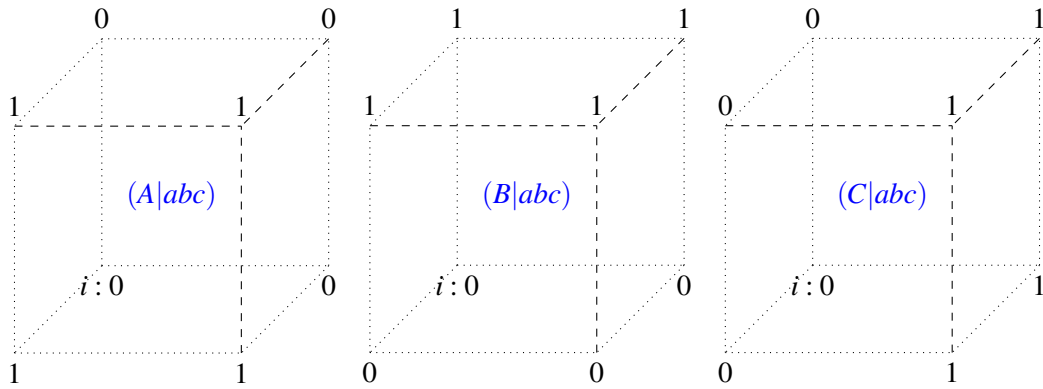


Figura 4.2: Descomposición del subsistema en tres n -cubos (tensores) representando las probabilidades condicionales de cada variable futura. Los valores asociados a cada vértice representan la probabilidad de que la variable correspondiente tome el valor 0 en $t+1$ dado el estado específico del subsistema en tiempo t .

Por ejemplo, para el estado inicial $abc_t = 000$:

- $P(A_{t+1} = 0 | abc_t = 000) = 0$ (tensor de A)
- $P(B_{t+1} = 0 | abc_t = 000) = 0$ (tensor de B)
- $P(C_{t+1} = 0 | abc_t = 000) = 0$ (tensor de C)

Esto implica que $P(ABC_{t+1} = 000 | abc_t = 000) = 0 \cdot 0 \cdot 0 = 0$, es decir, el subsistema permanece en el estado 000 con probabilidad 0.

4.2 Cálculo de Función de Costo y Exploración del Espacio

Habiendo establecido la representación geométrica y la descomposición tensorial del subsistema, procedemos ahora a aplicar la función de costo $t(i, j)$ para analizar las transiciones entre estados y construir la tabla \mathbb{T} que guiará la identificación de biparticiones óptimas.

4.2.1 Ejemplo de Transición: Estado 000 a Estado 011

Para ilustrar el proceso de cálculo de la función de costo, analizaremos en detalle la transición desde el estado inicial $i = 000$ hasta el estado $j = 011$.

Análisis de Caminos Posibles

La distancia de Hamming entre $i = 000$ y $j = 011$ es $d_H(000, 011) = 2$, lo que significa que difieren en exactamente dos posiciones (la segunda y tercera). Existen dos posibles caminos óptimos entre estos estados:

1. $000 \rightarrow 010 \rightarrow 011$ (cambiando primero B y luego C)
2. $000 \rightarrow 001 \rightarrow 011$ (cambiando primero C y luego B)

Cada uno de estos caminos tiene longitud 2 (mínima posible), pero pueden tener diferentes costos asociados dependiendo de las probabilidades condicionales en los estados intermedios.

4.2.2 Construcción Sistemática de la Tabla de Costos

Partiendo del estado inicial $i = 000$, aplicamos la función de costo $t(i, j)$ para calcular el costo de transición hacia todos los posibles estados j . Este proceso puede seguir una estrategia bottom-up, calculando primero los costos para estados adyacentes (distancia Hamming = 1), luego para estados a distancia 2, y finalmente para el estado más lejano (distancia 3).

La función de costo $t(i, j)$ se define correctamente como:

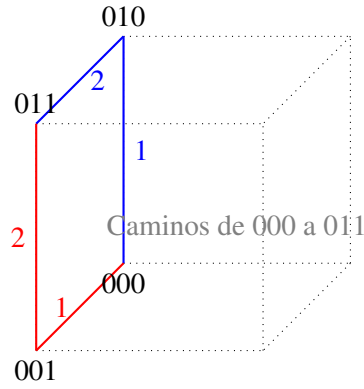


Figura 4.3: Visualización de los dos posibles caminos 'óptimos' desde el estado 000 hasta el estado 011. El camino azul pasa por 010, mientras que el camino rojo pasa por 001.

$$t_x(i, j) = \gamma \cdot \left(|X[i] - X[j]| + \sum_{k \in N(i, j)} \{t(k, j)\} \right) \quad (4.1)$$

donde:

- $\gamma = 2^{-d_H(i, j)}$ es el factor de decrecimiento exponencial
- $d_H(i, j)$ es la distancia de Hamming entre los estados i, j
- $X[s]$ es el valor de probabilidad asociado al estado s
- $N(i, j)$ son los sub-vecinos inmediatos de i que están en algún camino óptimo hacia j

Utilizando los valores de probabilidad de cada variable que se muestran en los tensores elementales:

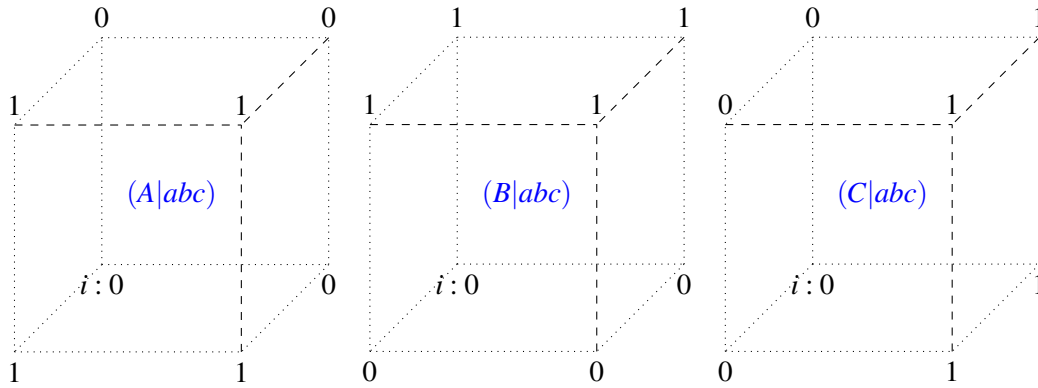


Figura 4.4: Descomposición del subsistema en tres n -cubos (tensores) representando las probabilidades condicionales de cada variable futura. Los valores asociados a cada vértice representan la probabilidad de que la variable correspondiente tome el valor indicado en $t + 1$ dado el estado específico del subsistema en tiempo t .

Cálculos para la Variable A

Procedemos a calcular sistemáticamente los costos de transición desde el estado inicial 000 hacia todos los demás estados para la variable A:

Para estados a distancia Hamming = 1:

$$t_A(000, 100) = 2^{-1} \cdot (|A[000] - A[100]| + 0) \quad (4.2)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.3)$$

$$t_A(000, 010) = 2^{-1} \cdot (|A[000] - A[010]| + 0) \quad (4.4)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.5)$$

$$t_A(000, 001) = 2^{-1} \cdot (|A[000] - A[001]| + 0) \quad (4.6)$$

$$= \frac{1}{2} \cdot (|0 - 1| + 0) = \frac{1}{2} \quad (4.7)$$

Para estados a distancia Hamming = 2:

$$t_A(000, 110) = 2^{-2} \cdot (|A[000] - A[110]| + t_A(000, 100) + t_A(000, 010)) \quad (4.8)$$

$$= \frac{1}{4} \cdot (|0 - 0| + 0 + 0) = 0 \quad (4.9)$$

$$t_A(000, 101) = 2^{-2} \cdot (|A[000] - A[101]| + t_A(000, 100) + t_A(000, 001)) \quad (4.10)$$

$$= \frac{1}{4} \cdot (|0 - 1| + 0 + \frac{1}{2}) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.11)$$

$$t_A(000, 011) = 2^{-2} \cdot (|A[000] - A[011]| + t_A(000, 010) + t_A(000, 001)) \quad (4.12)$$

$$= \frac{1}{4} \cdot (|0 - 1| + 0 + \frac{1}{2}) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.13)$$

Para el estado a distancia Hamming = 3:

$$t_A(000, 111) = 2^{-3} \cdot (|A[000] - A[111]| + t_A(000, 110) + t_A(000, 101) + t_A(000, 011)) \quad (4.14)$$

$$= \frac{1}{8} \cdot (|0 - 1| + 0 + \frac{3}{8} + \frac{3}{8}) = \frac{1}{8} \cdot (1 + \frac{3}{8} + \frac{3}{8}) = \frac{14}{64} \quad (4.15)$$

Cálculos para la Variable B

Realizamos los mismos cálculos para la variable B:

Para estados a distancia Hamming = 1:

$$t_B(000, 100) = 2^{-1} \cdot (|B[000] - B[100]| + 0) \quad (4.16)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.17)$$

$$t_B(000,010) = 2^{-1} \cdot (|B[000] - B[010]| + 0) \quad (4.18)$$

$$= \frac{1}{2} \cdot (|0 - 1| + 0) = \frac{1}{2} \quad (4.19)$$

$$t_B(000,001) = 2^{-1} \cdot (|B[000] - B[001]| + 0) \quad (4.20)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.21)$$

Para estados a distancia Hamming = 2:

$$t_B(000,110) = 2^{-2} \cdot (|B[000] - B[110]| + t_B(000,100) + t_B(000,010)) \quad (4.22)$$

$$= \frac{1}{4} \cdot (|0 - 1| + 0 + \frac{1}{2}) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.23)$$

$$t_B(000,101) = 2^{-2} \cdot (|B[000] - B[101]| + t_B(000,100) + t_B(000,001)) \quad (4.24)$$

$$= \frac{1}{4} \cdot (|0 - 0| + 0 + 0) = 0 \quad (4.25)$$

$$t_B(000,111) = 2^{-3} \cdot (|C[000] - C[111]| + t_B(000,110) + t_B(000,101) + t_B(000,011)) \quad (4.26)$$

$$= \frac{1}{4} \cdot (|0 - 1| + \frac{1}{2} + 0) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.27)$$

Para el estado a distancia Hamming = 3:

$$t_B(000,111) = 2^{-3} \cdot (|B[000] - B[111]| + t_B(000,110) + t_B(000,101) + t_B(000,011)) \quad (4.28)$$

$$x = \frac{1}{8} \cdot (|0 - 1| + \frac{3}{8} + 0 + \frac{3}{8}) = \frac{1}{8} \cdot (1 + \frac{3}{4}) = \frac{7}{32} \quad (4.29)$$

Cálculos para la Variable C

Finalmente, calculamos los costos para la variable C:

Para estados a distancia Hamming = 1:

$$t_C(000,100) = 2^{-1} \cdot (|C[000] - C[100]| + 0) \quad (4.30)$$

$$= \frac{1}{2} \cdot (|0 - 1| + 0) = \frac{1}{2} \quad (4.31)$$

$$t_C(000,010) = 2^{-1} \cdot (|C[000] - C[010]| + 0) \quad (4.32)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.33)$$

$$t_C(000,001) = 2^{-1} \cdot (|C[000] - C[001]| + 0) \quad (4.34)$$

$$= \frac{1}{2} \cdot (|0 - 0| + 0) = 0 \quad (4.35)$$

Para estados a distancia Hamming = 2:

$$t_C(000,110) = 2^{-2} \cdot (|C[000] - C[110]| + t_C(000,100) + t_C(000,010)) \quad (4.36)$$

$$= \frac{1}{4} \cdot (|0 - 1| + \frac{1}{2} + 0) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.37)$$

$$t_C(000,101) = 2^{-2} \cdot (|C[000] - C[101]| + t_C(000,100) + t_C(000,001)) \quad (4.38)$$

$$= \frac{1}{4} \cdot (|0 - 1| + \frac{1}{2} + 0) = \frac{1}{4} \cdot (1 + \frac{1}{2}) = \frac{3}{8} \quad (4.39)$$

$$t_C(000,011) = 2^{-2} \cdot (|C[000] - C[011]| + t_C(000,010) + t_C(000,001)) \quad (4.40)$$

$$= \frac{1}{4} \cdot (|0 - 0| + 0 + 0) = 0 \quad (4.41)$$

Para el estado a distancia Hamming = 3:

$$t_C(000,111) = 2^{-3} \cdot (|C[000] - C[111]| + t_C(000,110) + t_C(000,101) + t_C(000,011)) \quad (4.42)$$

$$= \frac{1}{8} \cdot (|0 - 1| + \frac{3}{8} + \frac{3}{8} + 0) = \frac{1}{8} \cdot (1 + \frac{3}{4}) = \frac{7}{32} \quad (4.43)$$

4.2.3 Resultados Completos de la Tabla de Costos

La tabla siguiente muestra los costos calculados para todas las transiciones posibles desde el estado inicial 000 para cada variable del sistema:

Transición	Variable A	Variable B	Variable C
$t(000,000)$	0	0	0
$t(000,100)$	0	0	0.5
$t(000,010)$	0	0.5	0
$t(000,110)$	0	0.375	0.375
$t(000,001)$	0.5	0	0
$t(000,101)$	0.375	0	0.375
$t(000,011)$	0.375	0.375	0
$t(000,111)$	0.219	0.219	0.219

Cuadro 4.2: Costos de transición calculados desde el estado 000 hacia todos los posibles estados para cada variable del sistema.

	A	B	C	D	E
1		000	001	010	011
2	000		0.50	0.50	1
3	001	0.50	1	1	1.50
4	010	0.50	1	0	0.50
5	011	1	1.50	0.50	1
6	100	0.50	0	1	0.50
7	101	1	0.50	1.50	1
8	110	1	0.50	0.50	0
9	111	1.50	1	1	0.50

Figura 4.5: Ejecución de biparticiones mediante fuerza bruta para validación computacional

4.2.4 Identificación de Biparticiones Óptimas

Analizando la tabla de transiciones \mathbb{T} , podemos identificar patrones de complementariedad que revelan biparticiones óptimas. Una bipartición óptima se caracteriza por tener costos de transición complementarios que minimizan la discrepancia total.

De la tabla de costos, identificamos las biparticiones mínimas (vistas en la tabla del análisis de complementariedad).


Bipartición	Transiciones A	Transiciones B	Transiciones C
Primera	$A(000, 100) = 0$	$B(000, 100) = 0$	$C(000, 011) = 0$
Segunda	$A(000, 010) = 0$	$B(000, 101) = 0$	$C(000, 010) = 0$
Tercera	$A(000, 110) = 0$	$B(000, 001) = 0$	$C(000, 001) = 0$

Cuadro 4.3: Biparticiones identificadas mediante análisis de complementariedad en la tabla de costos.

Estas biparticiones representan divisiones "*naturales*" del sistema en términos de la estructura causal revelada por el análisis topológico. La propiedad clave que observamos es que los estados complementarios forman biparticiones coherentes, con costos de transición que se complementan para minimizar la discrepancia global.

Es importante destacar que este análisis nos permite identificar directamente las biparticiones óptimas sin necesidad de evaluar exhaustivamente todas las posibles combinaciones, lo que representa una significativa reducción en la complejidad computacional del problema.

Un aspecto notable es la tendencia a favorecer biparticiones donde solo se marginaliza una variable, lo que es consistente con la observación empírica de que tales biparticiones suelen ser óptimas en muchos sistemas prácticos. Esto se debe a que el costo de movimiento en el espacio de estados es mínimo cuando se preserva la mayor parte de la estructura causal original.



5. Entregables y Expectativas

Este capítulo establece claramente los requerimientos, alcances y criterios de evaluación para la implementación del enfoque propuesto. El objetivo es proporcionar una guía concreta que clarifique el desarrollo de una solución eficiente y correcta al problema de bipartición óptima, aprovechando las propiedades topológicas del sistema.

La propuesta metodológica que hemos presentado se fundamenta en principios teóricos sólidos que permiten reducir significativamente la complejidad computacional del problema. Sin embargo, su implementación práctica requiere atención a numerosos detalles técnicos y de diseño para garantizar tanto la correctitud de los resultados como la eficiencia del proceso.

5.1 Componentes Esenciales de la Implementación

La implementación del enfoque geométrico debe mantener compatibilidad con los componentes existentes mientras incorpora las estructuras y algoritmos necesarios para explotar las propiedades topológicas del espacio de estados. A continuación, se detallan los componentes fundamentales requeridos.

5.1.1 Módulos de Software Requeridos

El proyecto debe implementarse como una extensión del framework existente, siguiendo una estructura modular que permita tanto la integración con los componentes actuales como la validación comparativa de resultados.

Estructura del Código Base

El código base del proyecto sigue una arquitectura orientada a objetos con un diseño basado en estrategias, donde diferentes algoritmos para identificar biparticiones óptimas se implementan como clases que heredan de una interfaz común. La estructura de directorios relevante es la siguiente:

```
src/  
  controllers/  
    strategies/
```

```

__init__.py
sia.py          # Clase base con la interfaz común SIA
qnodes.py      # Implementación existente (QNodes)
...
geometric.py    # NUEVA ESTRATEGIA (a implementar)
models/
...
tests/
  PruebasIniciales.xlsx # Conjunto de pruebas
  ...

```

La implementación del enfoque geométrico debe realizarse creando una nueva clase `GeometricSIA` en el archivo `src/controllers/strategies/geometric.py` que herede de la clase base `SIA`. Esta organización garantiza la interoperabilidad con el resto del sistema y facilita la comparación de resultados entre diferentes estrategias.

Implementación de la Clase `GeometricSIA`

La clase `GeometricSIA` debe implementar, como mínimo, los siguientes métodos:

```

class Geometric(SIA):
    def __init__(self, gestor):
        super().__init__(gestor)
        # ...atributos útiles a la estrategia

    def aplicar_estrategia(self):
        """
        Implementa el algoritmo para encontrar la bipartición óptima
        utilizando el enfoque topológico.
        """
        # 1. Construir la representación n-dimensional del sistema
        # 2. Calcular la tabla de costos (T) para cada variable
        # 3. Identificar las biparticiones candidatas
        # 4. Evaluar y seleccionar la bipartición óptima
        # 5. Retornar el resultado en formato compatible

```

El modelo matemático u algoritmo central puede resumirse en el siguiente pseudocódigo:

La función `CalcularCostoDeTransicion` implementa la fórmula recursiva:

$$t(i, j) = \gamma \cdot \left(|X[i] - X[j]| + \sum_{k \in N(i, j)} \{t(k, j)\} \right) \quad (5.1)$$

donde $\gamma = 2^{-d_H(i, j)}$ y $N(i, j)$ son los vecinos de i en caminos óptimos hacia j .

La complejidad computacional del algoritmo propuesto es:

- **Complejidad temporal:** $O(n \cdot 2^n)$ donde n es el número de variables, en contraste con $O(2^{2n-1})$ de la búsqueda exhaustiva tradicional.
- **Complejidad espacial:** $O(1)$ para almacenar la tabla completa de transiciones.

Posibles optimizaciones incluyen:

- Cálculo bajo demanda de los valores de transición, reduciendo la complejidad espacial.
- Paralelización del cálculo de la tabla \mathbb{T} por variables independientes.
- Explotación de simetrías para reducir el número de cálculos necesarios.

Algorithm 2 Algoritmo Geométrico

```

1: Input: Subsistema  $S$  con  $n$  variables
2: Output: Bipartición óptima  $B_{opt}$ 
3:  $tensors \leftarrow$  DescomponerEnTensores( $S$ )
4:  $T \leftarrow$  InicializarTablaDeTransiciones()
5: for cada variable  $v$  en  $S$  do
6:   for cada estado inicial  $i$  do
7:     for cada estado final  $j$  do
8:        $T[v, i, j] \leftarrow$  CalcularCostoDeTransicion( $i, j, tensors[v]$ )
9:     end for
10:   end for
11: end for
12:  $candidates \leftarrow$  IdentificarBiparticionesCandidatas( $T$ )
13:  $B_{opt} \leftarrow$  EvaluarCandidatos( $candidates, S, T$ )
14: return  $B_{opt}$ 

```

5.1.2 Interfaces y Estructuras de Datos

La implementación debe utilizar estructuras de datos optimizadas para representar eficientemente el espacio n -dimensional y realizar las operaciones requeridas por el algoritmo.

Especificación de Interfaces Principales

La interfaz principal que debe implementarse es la definida por la clase base SIA.

Posiblemente, se deben implementar métodos auxiliares específicos para el nuevo enfoque:

- `calcular_transicion_coste(...)`: Calcula el costo de transición entre estados.
- `identificar_candidatos(...)`: Identifica biparticiones candidatas basadas en \mathbb{T} .

Estructuras de Datos Optimizadas

Para una implementación eficiente, se recomiendan las siguientes estructuras de datos:

- **Caché de cálculos:** Utilizar tablas de memorización para valores de transición calculados previamente, evitando recálculos en la función recursiva.
- **Biparticiones candidatas:** Representar como pares de conjuntos de índices de variables.
- **Tabla de transiciones:** Implementar como diccionario anidado para acceso eficiente por coordenadas, o como matriz n -dimensional sparse si la dimensionalidad ya es muy alta.

El uso de estructuras optimizadas es crítico para el rendimiento del algoritmo, especialmente para sistemas con más de 25 variables, donde la dimensionalidad del espacio de estados crece exponencialmente.

5.2 Criterios de Evaluación y Casos de Prueba

La evaluación de la implementación se basará en dos aspectos fundamentales: la correctitud de los resultados (precisión) y el rendimiento computacional (eficiencia). A continuación, se detallan los criterios y métodos para esta evaluación.

5.2.1 Conjunto de Datos de Prueba

La implementación será evaluada utilizando el conjunto de pruebas predefinido en el archivo `tests/PruebasIniciales.xlsx`, que contiene una colección de sistemas de prueba con diferentes características y complejidades.

Utilización del Conjunto de Pruebas Existente

El conjunto de pruebas incluye sistemas con:

- Diferentes números de variables (desde 10 hasta 20)
- Diversas estructuras causales (desde redes deterministas hasta estocásticas)

Para cada sistema de prueba, se proporciona tanto la matriz de transición (TPM) como la bipartición óptima esperada, calculada mediante el algoritmo exhaustivo de PyPhi.

Estrategias de Validación Cruzada

La validación de la implementación debe seguir estas etapas:

1. **Validación por casos pequeños:** Verificar la correctitud en sistemas pequeños (3-5 variables) donde el resultado óptimo se conoce y puede verificarse manualmente.
2. **Comparación con PyPhi:** Para sistemas de tamaño medio (6-10 variables), comparar los resultados con los obtenidos por PyPhi, que implementa el algoritmo exhaustivo y garantiza resultados óptimos.
3. **Escalabilidad:** Para sistemas grandes (>20 variables), donde reitero PyPhi puede ser impracticable, evaluar la consistencia interna y el comportamiento asintótico del algoritmo.

PyPhi proporciona una referencia "*standar*" para la validación, ya que implementa el algoritmo exhaustivo que garantiza encontrar la bipartición óptima. Sin embargo, su enfoque de fuerza bruta limita su aplicabilidad a sistemas de tamaño moderado.

5.2.2 Métricas de Desempeño

La evaluación cuantitativa de la implementación se basará en las siguientes métricas:

Análisis de Error y Precisión

Dado que el enfoque geométrico puede no garantizar siempre la bipartición absolutamente óptima (especialmente en sistemas complejos), es crucial cuantificar la calidad de las soluciones encontradas:

- **Tasa de acierto exacto:** Porcentaje de casos donde la bipartición encontrada coincide exactamente con la óptima (según PyPhi).
- **Error relativo en Φ :** Para casos donde la bipartición difiere, calcular:

$$E_{rel} = \frac{|\Phi_{\text{óptimo}} - \Phi_{\text{encontrado}}|}{\Phi_{\text{óptimo}}} \quad (5.2)$$

- **Distancia estructural:** Medida de similitud entre la bipartición encontrada y la óptima, utilizando métricas como la distancia de Jaccard entre los conjuntos que forman las particiones.

La Tabla 5.1 establece los umbrales de aceptabilidad para estas métricas:

Nivel de calidad	Tasa de acierto	Error relativo máx.	Dist. estructural máx.
Excelente	>90 %	<1 %	<0.1
Bueno	>80 %	<5 %	<0.2
Aceptable	>70 %	<10 %	<0.3
Insuficiente	<70 %	>10 %	>0.3

Cuadro 5.1: Umbrales de aceptabilidad para las métricas de error y precisión.

Comparativa de Rendimiento Temporal

El rendimiento computacional se evaluará mediante:

- **Speedup relativo:** Factor de aceleración respecto a PyPhi, calculado como:

$$S_{rel} = \frac{T_{\text{PyPhi}}}{T_{\text{Geometric}}} \quad (5.3)$$

- **Escalabilidad:** Relación entre el tiempo de ejecución y el tamaño del sistema, comparando el comportamiento asintótico observado con el teórico ($O(n \cdot 2^n)$).
- **Uso de memoria:** Consumo máximo de memoria durante la ejecución del algoritmo.

La implementación debe lograr un speedup significativo respecto a PyPhi para sistemas de tamaño moderado (>10 variables) y permitir el análisis de sistemas grandes (>20 variables) que serían inabordables con el enfoque exhaustivo.

En síntesis, una implementación fina debe lograr un equilibrio entre precisión y eficiencia, proporcionando resultados de alta calidad con una reducción significativa en la complejidad computacional temporal respecto a enfoques exhaustivos tradicionales.



6. Perspectivas y Desarrollos Futuros

Este capítulo explora posibles direcciones para extender y mejorar el enfoque geométrico propuesto. Más que una parte estricta de la guía, las ideas representan oportunidades de exploración e innovación que pueden enriquecer la metodología y que ampliar la aplicabilidad a sistemas más complejos o características '*particulares*'.

La naturaleza del problema de la Mínima Pérdida de Información (MIP) y su representación geométrica ofrece un terreno para desarrollos tanto teóricos como prácticos. A continuación, se presentan algunas áreas prometedoras para investigación futura y refinamiento de la metodología.

6.1 Extensiones Teóricas del Enfoque Geométrico

El marco teórico presentado en este documento establece las bases, pero existen numerosas direcciones en las que esta teoría podría extenderse y refinarse.

Refinamiento de la Función de Costo

La función de costo $t(i, j)$ propuesta utiliza un factor de decrecimiento exponencial $\gamma = 2^{-d_H(i, j)}$ para ponderar la influencia de las transiciones entre estados. Esta elección ofrece resultados prometedores, pero modificaciones podrían mejorar la capacidad para identificar estructuras causales:

- **Factores de decrecimiento alternativos:** Explorar otros patrones de decrecimiento como funciones polinomiales, logarítmicas o incluso adaptativas que se ajusten a las características específicas del sistema.
- **Ponderación por relevancia causal:** Incorporar información sobre la fuerza de las relaciones causales entre variables específicas, permitiendo que ciertas dimensiones del hipercubo tengan mayor influencia en el costo de transición.
- **Métricas de distancia alternativas:** Investigar el uso de otras métricas además de la distancia de Hamming, como distancias basadas en información, que podrían capturar mejor ciertos aspectos de la estructura causal.
- **Ajuste dinámico de parámetros:** Desarrollar métodos para calibrar automáticamente los parámetros de la función de costo basándose en las características observadas del sistema.

Estas modificaciones podrían mejorar significativamente la precisión del método, especialmente en sistemas con estructuras causales complejas o atípicas.

Extensión a Sistemas de Mayor Dimensionalidad

Aplicar el enfoque geométrico a sistemas con un gran número de variables podría simplificarse con:

- **Técnicas de reducción dimensional:** Desarrollar métodos para identificar y eliminar dimensiones redundantes o menos relevantes del hipercubo, simplificando el análisis sin comprometer significativamente la precisión.
- **Análisis jerárquico:** Implementar un enfoque multinivel donde primero se realizan particiones gruesas del sistema en componentes principales, y luego se refina el análisis dentro de cada componente.
- **Métodos de muestreo inteligente:** Para sistemas extremadamente grandes, explorar técnicas de muestreo del espacio de estados que prioricen regiones con mayor relevancia causal.
- **Aproximaciones en espacios continuos:** Investigar representaciones continuas aproximadas del espacio discreto de estados, permitiendo aplicar técnicas del análisis matemático y la geometría diferencial.

Estas extensiones podrían hacer viable el análisis de sistemas con decenas o incluso cientos de variables, abriendo nuevas posibilidades para aplicaciones en biología, neurociencia, economía y otras disciplinas donde los sistemas de interés suelen tener alta dimensionalidad.

6.2 Innovaciones Algorítmicas y Computacionales

Más allá de las extensiones teóricas, existen numerosas oportunidades para mejorar los aspectos algorítmicos y computacionales de la metodología.

6.2.1 Optimización Heurística Basada en Propiedades Topológicas

La tabla de costos \mathbb{T} generada por nuestro análisis contiene patrones estructurales que podrían explotarse mediante heurísticas especializadas:

- **Algoritmos genéticos adaptados:** Diseñar operadores de cruce y mutación que aprovechen la estructura topológica del hipercubo para explorar de manera más eficiente el espacio de biparticiones.
- **Búsqueda tabú con memoria topológica:** Implementar estrategias de búsqueda local que utilicen la información de adyacencia en el hipercubo para guiar la exploración y evitar óptimos locales.
- **Optimización por enjambre de partículas:** Adaptar algoritmos PSO para explorar el espacio de biparticiones utilizando la distancia de Hamming como métrica natural.
- **Aprendizaje por refuerzo:** Entrenar agentes que aprendan a identificar patrones en la tabla de costos que correlacionen con biparticiones de alta calidad.

Estas heurísticas podrían permitir identificar biparticiones cuasi-óptimas con una fracción del costo computacional del análisis exhaustivo.

6.2.2 Implementación Eficiente y Escalable

Para maximizar el rendimiento práctico del enfoque, se pueden considerar diversas optimizaciones técnicas:

- **Paralelización adaptativa:** Desarrollar esquemas de paralelización que se ajusten dinámicamente a la estructura del problema y a la arquitectura computacional disponible.
- **Computación en GPU:** Implementar algoritmos específicos para GPUs que exploten el paralelismo masivo en el cálculo de la tabla de costos.

- **Estructuras de datos probabilísticas:** Utilizar estructuras como filtros de Bloom o Count-Min Sketch para aproximar la tabla de costos con menor consumo de memoria.
- **Modelos predictivos:** Entrenar algún que otro modelo de aprendizaje automático que prediga las biparticiones óptimas basándose en características del sistema, evitando el cálculo completo de la tabla de costos.

Estas optimizaciones podrían permitir aplicar el enfoque a problemas de escala industrial o científica que actualmente están fuera del alcance de las metodologías existentes.

En resumen, aunque la metodología propuesta representa un avance significativo en el análisis de biparticiones óptimas, existe un vasto potencial para refinamientos, extensiones y aplicaciones innovadoras libres para tí, gran pensador@. Estas direcciones futuras no solo prometen mejorar la eficiencia y precisión del método, sino también expandir su aplicabilidad dominios y problemas de mayor complejidad.