



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – CONTROL

**MODELADO DE SISTEMAS DISTRIBUIDOS MÓVILES CONSIDERANDO
RESTRICCIONES DE TIEMPO REAL**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
JOSE EDUARDO VILLA HERRERA

TUTOR PRINCIPAL
HÉCTOR BENÍTEZ PÉREZ
**INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN
SISTEMAS, UNAM**

CIUDAD DE MÉXICO, ENERO 2020



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Arteaga Pérez Marco Antonio
Secretario: Dr. Álvarez Icaza Longoria Luis A.
1 er. Vocal: Dr. Benítez Pérez Héctor
2 do. Vocal: Dr. Espinosa Pérez Gerardo René
3 er. Vocal: Dr. Ortega Arjona Jorge

Lugar donde se realizó la tesis: Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Ciudad Universitaria.

TUTOR DE TESIS:

Dr. Héctor Benítez Pérez

FIRMA

DEDICATORIA

- *A mi madre de corazón Tomasa Herrera Bautista: en vida me formaste con los principios y valores causantes del ser humano que soy hoy día, así como fuiste mi motor en todo desafío de mi vida. Por eso y más, siempre estarás presente espiritualmente conmigo.*
- *A mi madre biológica Luz del Carmen Villa, así como a mi padre de corazón Jesús Paniagua: a distancia no me han dejado de apoyar día a día en esta turbulenta pero enriquecedora etapa de mi vida.*
- *A mi hermana y amiga, Montserrath Villa: en vida tuviste esa forma particular de apoyarme. Quisiera haberte apoyado de mejor forma en tu formación, sin embargo el tiempo y las circunstancias no me perdonaron; pero tu sí. ¡Te amo!*
- *A mis padrinos de toda la vida, Isabel y Ciro: quienes son mis padres también; así como a Maritza, Carlos y Francisco: fuentes de apoyo y fortaleza en todo momento.*
- *A mi amigo y formador académico Dr. Saúl de la Rosa Nieves: valoro y aprecio su metodología y conocimiento que ha influido en el buen ingeniero que soy.*

Agradecimientos

- A la Universidad Nacional Autónoma de México: el privilegio de ser universitario de la máxima casa de estudios de América Latina es de las experiencias más gratas de mi vida.
- Al Consejo Nacional de Ciencia y Tecnología (*CONACyT*): por el apoyo económico brindado durante mi formación como Maestro en Ingeniería Eléctrica (CVU 855737).
- Al Dr. Héctor Benítez Pérez, tutor de mi tesis y director del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (*IIMAS*): por su paciencia, pero sobre todo, por su dedicación y compromiso en formar a mi y a todo su equipo de trabajo en pro de la generación de conocimiento científico-tecnológico.
- A mis sinodales: Dr. Marco Arteaga, Dr. Luis Álvarez Icaza, Dr. Jorge Ortega y Dr. Gerardo Espinosa. Por su voto de confianza en mi trabajo de tesis. Mención especial para el Dr. Jorge, por sus valiosas observaciones que enriquecieron mi trabajo; me hubiera gustado conocerle antes.
- Al grupo de trabajo del Dr. Héctor Benítez (Laboratorio de Control y DISCA IIMAS): Nora Padilla, José Ángel Hermosillo, José Alberto Aparicio, Alejandro Velázquez, Gerardo Castillo, Oscar Arana, Oriol Castillo. Por estos dos años y medio de compañerismo, hermandad e intercambio de conocimiento.
- Al área secretarial de la Dirección del IIMAS: Violeta y Teresa. Siempre atentas para apoyarme en tiempo y forma con el papeleo y logística durante nuestras juntas, seminarios y para mi examen profesional.

INTRODUCCIÓN

Un sistema distribuido es un conjunto de entidades computacionales secuenciales llamados *procesos*, los cuales trabajan conjuntamente para resolver algún problema denominado *tarea*, considerando su complejidad, fallas y tiempos. Los procesos pueden interactuar mediante algún mecanismo, como el envío-recepción de mensajes y/o el compartimiento de memoria. Además, cada proceso ejecuta un programa que define como y cuando se da la comunicación con otros procesos; dichos programas en conjunto conforman lo que se conoce como un *protocolo* o *algoritmo distribuido*.

Un *sistema distribuido móvil (SDM)* se toma caso de estudio para esta tesis. El SDM se compone de dispositivos individuales llamados *nodos*, los cuales cuentan principalmente con procesador y memoria locales, así como una fuente de potencia eléctrica. Cada nodo se comunica por el envío-recepción de mensajes con sus nodos vecinos de forma inalámbrica, y en general, sin dependencia de alguna infraestructura física dada. La información fundamental a ser comunicada por cada nodo comprende: su posición geográfica relativa y su estado. Para el modelado de la comunicación de los nodos, la comunidad de sistemas distribuidos ha implementado enfoques determinísticos y aleatorios de forma individual o colectiva, apoyados principalmente con la teoría de gráficas y la capacidad cinética del problema de agrupamiento de nodos (*clustering*).

El enfoque para el análisis y modelado de SDM's aplicado en este trabajo se fundamenta en la *topología*, la cual es la matemática que estudia las propiedades esenciales de los espacios abstractos, así como las transformaciones entre ellos [14]. La abstracción de un espacio X requiere fundamentalmente de un conjunto, así como una noción de proximidad expresada como un sistema de subconjuntos abierto, donde el especificar una métrica es permisible, pero no requerido. Las redes de sensores, grafos, conjuntos de datos, señales, procesamiento de imágenes, son interpretados como espacios topológicos de aplicación. [14] Para un SDM se consideran las siguientes propiedades esenciales:

- **Views (vistas) o estados locales.**- Esta propiedad es individual por proceso, y representa una vista local de su estado presente con su entorno. Por lo tanto, no se tiene certeza de las vistas de otros procesos.
- **Evolución de las vistas locales.**- Por la comunicación entre procesos, es posible modificar sus vistas locales. Sin embargo, esta comunicación puede presentar retrasos, fallas, o simplemente comunicación parcial.

Así, la idea principal del enfoque trabajado es la representación de todas las posibles vistas locales de los procesos en un determinado tiempo, mediante la construcción de invariantes topológicos de espacios y mapeos entre ellos por medio de objetos geométricos combinatoriales llamados *simplicial complexes*, los cuales son insensibles a cambios de coordenadas y a deformaciones.

Problemática

La nueva era digital involucra trabajar con redes compuestas por cientos o miles de nodos para la realización de una o varias tareas. Ya que los SDM's son inherentemente asíncronos, sus procesos pueden demorarse sin previo aviso por una variedad de razones, incluidas las interrupciones, la saturación de memoria, los retrasos en la comunicación para la planificación de tareas y fallas en la infraestructura. Esto ha provocado un gran interés en el desarrollo de herramientas analíticas para predecir su comportamiento, así como para controlar las complejidades de tales sistemas a gran escala. El diseño de algoritmos para el despliegue, la localización, el ciclo de trabajo, la comunicación y la verificación de cobertura en tiempo real de SDM's forman el núcleo de esta área activa de investigación.

Objetivos

0.0.1. Objetivo general

Modelar un SDM considerando restricciones de tiempo real, con base en la topología combinacional y algebraica, ya que con esta teoría se pueden estimar características globales (estados) del SDM con solo tener información local; es decir, considerando la información conjunta de k nodos y sus nodos vecinos.

Los elementos a ser considerados para el modelo a desarrollar son:

- Población.
- Topología de interconexión.
- Movilidad de los nodos.
- Construcción de grupos.
- Relación entre la cercanía de grupos.

0.0.2. Objetivos particulares

- Establecer límites inferiores para los problemas de cardinalidad de los cúmulos, la cual es una condición natural para asegurar la asignación adecuada de los recursos.
- Además, al trabajar con espacios topológicos y sus invariantes topológicos, se dará una estimación al problema de cobertura de un SDM, cuando los nodos carecen de información geográfica (localización y/u orientación).

Metodología

1. Trabajo relacionado.- Se investigarán en revistas de control y sistemas distribuidos los principales enfoques seguidos para el modelado de SDM's, para así evaluar sus alcances y limitaciones a ser tomados en cuenta para el modelo a desarrollar.
2. Investigación del marco teórico.- Se resumirá el enfoque de la topología algebraica-combinacional, considerando los requerimientos fijados para el modelado del SDM a desarrollar.

3. Exploración de la infraestructura del SDM experimental con que se trabajará.- Dentro del laboratorio de sistemas de cómputo móvil del IIMAS se cuenta ya con tal planta, la cual está compuesta por los siguientes nodos: Arduino One con escudos Ethernet y otros con escudos WiFi, Arduino Yun, así como Raspberry Pi. Esta planta ya fue implementada por la doctora en ciencias Magali Arellano Vázquez, en la cual puso en marcha su algoritmo de encaminamiento por consenso, como parte de su tesis doctoral.
4. Propuestas de modelado del SDM a desarrollar.- Con todo lo anterior, se analizarán casos de estudio particulares con diversas distribuciones de nodos, para así llegar a un esquema generalizado del modelo del SDM considerando restricciones de tiempo.
5. Análisis del modelo propuesto:
 - a) Evaluación de las condiciones topológicas combinatorias del SDM que aseguren la existencia de un protocolo para la realización de una tarea. Se aterriza en las condiciones para la realización de un protocolo de acuerdo.
 - b) Evaluación de las condiciones para la cobertura y encaminamiento de los nodos.
 - c) Se dará un algoritmo general que aplique el modelo propuesto.

Estructura de la tesis

El presente proyecto se divide en 5 capítulos.

- El primer capítulo resume el trabajo relacionado sobre el modelado de SDM's, con el propósito de comparar sus alcances y limitaciones.
- En el segundo capítulo, se muestra el marco teórico fundamental sobre el modelado sistemas distribuidos bajo el enfoque topológico (combinatorio y homológico-algebraico).
- Para el tercer capítulo, se detalla el proceso de análisis para el modelado del SDM considerando restricciones de tiempo.
- En el cuarto capítulo se propone un algoritmo general para implementar el modelo desarrollado en la infraestructura de un SDM con el que cuenta el laboratorio de sistemas de cómputo móvil del IIMAS, la cual se compone de dieciséis nodos Arduino (One y Yun) y diez nodos Raspberry Pi.
- Por último, en el quinto capítulo se listan las conclusiones obtenidas del desarrollo de este nuevo modelo de un SDM, listando los aportes al conocimiento que realiza la presente tesis a la comunidad de control y de sistemas distribuidos; así como el trabajo a futuro propuesto.

Índice general

0.0.1. Objetivo general	4
0.0.2. Objetivos particulares	4
1. TRABAJO RELACIONADO	1
1.1. Panorama general de modelado de sistemas distribuidos móviles	1
1.2. Espacios topológicos y sus invariantes para el modelado de sistemas distribuidos	2
1.3. Cobertura y ruteo de SDM's	3
1.4. Redes temporales	4
1.5. Agrupamiento de nodos (clustering)	4
2. MARCO TEÓRICO	7
2.1. Enfoque topológico de un SDM	7
2.1.1. Elementos de la topología combinatorial	7
2.2. Modelos de sistemas con memoria compartida	9
2.2.1. Los snapshots inmediatos como interacción para la resolución de tareas	9
2.2.2. Forma operacional de sistemas con memoria compartida	9
2.2.3. Modelo combinatorial	12
2.3. Enfoque homológico de un SDM	15
2.3.1. Problemas de cobertura	16
2.3.2. Criterio homológico fundamental de cobertura	16
2.4. Comportamiento dinámico en tiempo real	17
3. ANÁLISIS PARA EL MODELO DESARROLLADO	19
3.1. Representación general del SDM	19
3.1.1. Marcos y consideraciones de operación	19
3.1.2. Cómputo descentralizado para la estimación de invariantes topológicos	21
3.2. Modelo de protocolos que resuelven tareas	21
3.2.1. Paso de mensajes entre nodos del SDM	22
3.3. Modelo de cobertura e identificación de cúmulos por simplicial complexes	25
3.3.1. Construcción de una representación local de \mathcal{R}	26
3.3.2. Cálculo de grupos homológicos mediante operadores Laplacianos	29
3.4. Protocolo de acuerdo	33
3.4.1. Caso estático	33
3.5. Restricciones para la comunicación en tiempo real	38
3.5.1. Comportamiento dinámico en tiempo real	38

4. ALGORITMO GENERAL PARA EL SDM DESARROLLADO	41
4.1. Algoritmo general propuesto para al SDM considerando RTT	41
4.1.1. Restricciones temporales	41
4.1.2. Flujo general del algoritmo	41
4.1.3. Subproceso pasoMensajes	41
4.1.4. Subproceso generarTuplas	42
4.1.5. Subproceso generarRips	43
4.1.6. Subproceso evaluarCobertura	43
4.1.7. Subproceso generarConsenso	44
4.2. Algoritmo a nivel nodal orientado a eventos	45
5. CONCLUSIONES Y TRABAJO A FUTURO	47
5.1. Conclusiones	47
5.2. Trabajo a futuro	47
A. Base de la topología combinacional: complexes	49
A.1. Construcciones comunes de los simplicial complex	50
A.2. Mapeos	52
A.3. Configuraciones aplicadas de los simplicial complexes	52
A.3.1. Complexes de estrategia e incertidumbres	53
A.3.2. Espacios de configuración de gráficos discretizados	53
A.3.3. Complexes de estados	54
B. Homología	55
B.0.1. Generalización de los conceptos de adyacencia y grado a partir de la teoría de grafos estándar	55
B.0.2. Cadenas complexes como objetos fundamentales de la homología	56
B.0.3. Características de la homología	58
B.0.4. Homología simplicial	59
B.0.5. Homología relativa	61
B.0.6. Mapas entre categorías: functorialidad	62
B.0.7. Complexes en homología	62
B.0.8. Homología persistente	63
Bibliografía	65

Capítulo 1

TRABAJO RELACIONADO

En este primer capítulo se hace una breve descripción de los enfoques seguidos por la comunidad de sistemas distribuidos para el modelado de tales sistemas. Para estos enfoques, se destacan algunos trabajos realizados, los cuales fueron considerados como base para el modelo a desarrollar en esta tesis.

Primeramente se presenta un panorama general de las metodologías (clasificación, enfoques, áreas de oportunidad y algoritmos) para modelar SDM's.

Además, se particulariza en algunos trabajos que atacan los problemas de cobertura y agrupamiento, así como el modelado sobre redes temporales; esto debido a que estos problemas y enfoques se derivan ideas para cubrir los objetivos de esta tesis.

1.1. Panorama general de modelado de sistemas distribuidos móviles

En [8, cap. 1], Herlihy, Kozlov y Rajsbaum conceptualizan a un sistema como una colección de procesos ligados por alguna forma de comunicación, y desarrollan los modelos de computación distribuida mostrados en la figura 1.1:

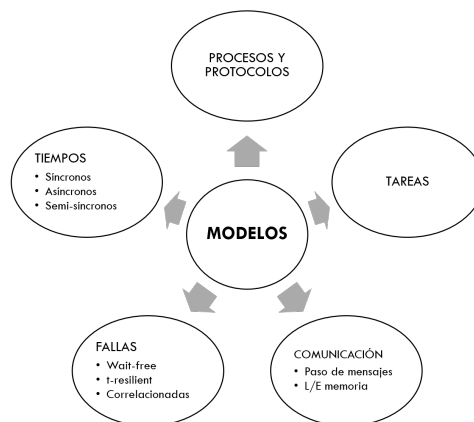


Figura 1.1: Modelos de computación distribuida.

Relacionado a [8], en [13, pag. 41] se interpreta como SDM's aquellos que se modelan por el paso de mensajes-asíncronos, e involucran internet y redes de sensores.

Por otro lado, en [20], Mohamed, Hamza y Saroit trabajan SDM's como redes de sensores inalámbricos móviles (M-WSN), las cuales pueden ser clasificadas de acuerdo al tipo de nodos desplegados, la estrategia de despliegue, la arquitectura de la red, la movilidad de nodos, el tipo de cobertura necesaria, y el modelo de sensado. En la figura 1.2 se muestra un esquema con la clasificación de M-WSN's presentado en este trabajo.

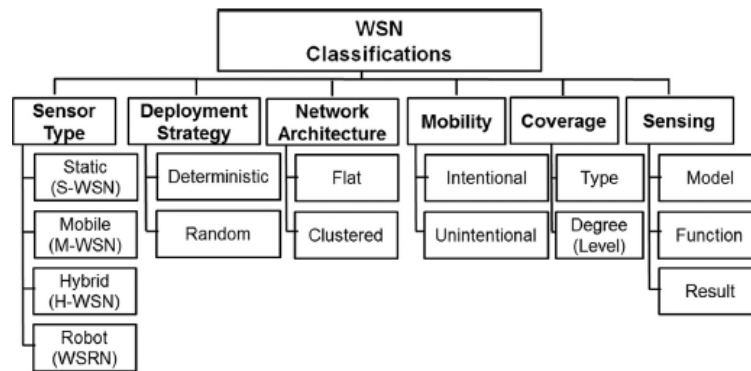


Figura 1.2: Taxonomía de M-WSM. Figura tomada de [20].

Una extensión de sistemas distribuidos considerando cómputo móvil es presentada en [3] por Benítez, Ortega, Méndez, Rubio y Esquivel. En este trabajo presentan algoritmos de planificación para obtener buena aproximación a retardos de tiempo acotados, considerando el gasto de tiempo en todas las etapas de comunicación y cómputo; así como los algoritmos de ruteo y consenso más importantes. Además, se muestra una metodología de diseño de red distribuida computacional, considerando características de tiempo real, planificadores, manejadores de tareas, prioridades, precedencias y consenso.

1.2. Espacios topológicos y sus invariantes para el modelado de sistemas distribuidos

Como ya se había hecho mención, la propiedades locales relacionadas por la interacción de los agentes de una red pueden ser capturadas por ciertos espacios topológicos. Estos espacios son comúnmente combinatorios, ya que son una generalización de modelos por grafos, además de que las propiedades globales de las características de la red corresponden a ciertos invariantes topológicos de estos espacios. Como ejemplos de propuestas sólidas de la aplicación de este enfoque se tienen:

- Problemas de cobertura para redes de sensores, considerando poca o nula información geográfica. Destacan los trabajos de V. de Silva, R. Ghrist y A. Muhammad. La computación trabajada ha sido por medio de grupos homológicos, ya sea en forma centralizada o distribuida.
- Modelado de concurrencia en sistemas distribuidos síncronos y asíncronos. Los grupos de trabajo M. Herlihy, S. Rajsbaum y N. Shavit han trabajado en conjunto y de forma separada en esta temática.

1.3. Cobertura y ruteo de SDM's

La cobertura es una característica de la calidad del servicio que brinda el sistema, y se refiere a que tan bien cada punto del área geográfica de interés es observada por los nodos desplegados.

Primeramente, un panorama general de los enfoques actuales para la cobertura son presentados en [20]. Se particulariza en redes de sensores inalámbricos móviles (M-WSN), mostrando en la figura 1.3 una clasificación de varios técnicas y algoritmos usados en M-WSN's para mejorar la cobertura después de despliegue inicial de los nodos, además para el mantenimiento de la cobertura ante fallas de nodos.



Figura 1.3: Enfoques para el problema de cobertura móvil en WSN. Figura tomada de [20].

Por otro lado, en [7] y [24], De Silva y Ghrist abordan este problema con bases de la homología algebraica persistente, cuya teoría la aplican sobre redes de sensores con mínimas posibilidades de geolocalización y sin consideraciones probabilistas. En particular, demuestran que una colección estacionaria de nodos pueden verificar cobertura dentro de un dominio acotado con topología de interconexión desconocida, siempre que el límite no esté demasiado contraído. Los escenarios que cubren son: cobertura plana estática, cobertura de barreras, así como cobertura de barrido dependiente del tiempo.

En [5], Chintakunta y Krim también se fundamentan en la homología algebraica para presentar un marco de referencia general para el análisis de redes de sensores, así como algoritmos para dar solución a la detección de huecos en la cobertura y localización, y detección de ataque de agujero de gusano.

Como caso particular de SDM's, la convergencia y cobertura sobre grafos para robots es trabajada en [4], Castañeda, Rajsbaum y Roy. Aquí, un conjunto de robots asíncronos son modelados como un *grafo de convergencia* G . Se considera que estos robots pueden fallar por choques entre ellos, que inician su movimiento desde posiciones no conocidas en un espacio dado, y que se pueden mover hacia posiciones cercanas entre ellos. En este trabajo se caracterizan los grafos en que los problemas

de acuerdo y cobertura por bordes son solucionables, así como se presentan algoritmos temporales óptimos para tales soluciones.

1.4. Redes temporales

Una red temporal es abstraída mediante un grafo $G(V, E)$, donde cada uno de sus e bordes tiene una *etiqueta temporal* $\lambda(e)$ que especifica el tiempo en que sus puntos finales (vértices) se comunican. Así, se puede ver a una red temporal como al par (G, λ) , donde λ es un mapa $\lambda : E \mapsto \mathbb{R}$.

Dentro de G , una *trayectoria temporal* P se usa para representar la noción del flujo de información, cuyas etiquetas son acorde al orden temporal.

En [17], Kempe motiva su trabajo al plantear la siguiente pregunta: ¿de que forma cambian las bases combinatorias y las propiedades de los grafos cuando se imponen adicionalmente condiciones temporales? Así, se dan resultados para dos tipos de problemas:

- **Conectividad.**- Se buscan trayectorias temporales para entre pares de nodos, conforme a la siguiente premisa: suponer que un nodo t ha aprendido parte de la información originalmente propiedad de un nodo s . Si se es consiente que algunos nodos pueden fallar: para algún valor de k , ¿hay k rutas internas con respecto al tiempo disjuntas entre nodos de s a t ? Para esto, en este trabajo se presenta una caracterización no trivial del teorema de Menger, en la cual se adapta para involucrar etiquetas temporales; así como se presenta un algoritmo (N-P completo) para la conectividad de grafos de Menger.
- **Inferencia.**- Se busca reconstruir (inferir) un etiquetado de tiempo parcialmente especificado de una red, siguiendo un historial observado del flujo de información.

1.5. Agrupamiento de nodos (clustering)

En [23], Chien Chun, Ni, Su, Zhengyu, Gao, Jie y Xianfeng estudian la capacidad cinética del problema de clustering (agrupamiento), el cual lo definen como sigue: “*dado un conjunto de k estaciones base con sus restricciones de capacidad de operación, así como un conjunto de n terminales, encontrar una asignación de terminales hacia las estaciones base tal que ninguna de estas opere más allá de su límite de capacidad, y que la suma de distancias al cuadrado entre terminales y estaciones base sea minimizada*”.

Wang y Hung formulan en [26] restricciones métricas de rendimiento dentro de su algoritmo de clustering. Su planta de control se desarrolla dentro de una red *MANET* (red Ad Hoc móvil, por sus siglas del inglés), la cual es una red inalámbrica de dispositivos móviles autoconfigurable y de poco infraestructura. Se considera que MANET es organizada por clusters (cúmulos), cada uno con un nodo líder llamado *clusterhead*, y el resto de nodos son miembros. Consideran además el consumo de potencia de un nodo clusterhead v , que es proporcional a la suma de distancias entre v y sus nodos miembros v' .

Por último, Yang y Li presentan en [27] su algoritmo de agrupamiento ponderado (WCA, por sus siglas del inglés) sobre una arquitectura MANET, el cual permite elegir un nodo principal de acuerdo al mapeo aperiódico de ponderaciones (pesos) de nodos, por lo que se puede mantener la estabilidad de la red, así como reducir el costo computacional y de comunicación.

Capítulo 2

MARCO TEÓRICO

En este segundo capítulo se exponen las bases para el modelado operacional de un SDM, considerando tareas, protocolos y fallas para un tiempo determinado. Ya con estos fundamentos, se presentan los elementos a implementar de la topología combinacional para desarrollo de un modelo con comunicación entre procesos, mediante la lectura-escritura sobre memoria compartida. Las definiciones, teoremas y lemas presentados son una compilación de ideas tomadas de [14], [16], [15] y [22].

Por otra parte, para solucionar el problema de cobertura y agrupamiento de los nodos del SDM, se aplica la *homología simplicial*, la cual es una construcción algebraica-topológica que determina características globales (ej. dominio de cobertura, fronteras, huecos) en un simplicial complex (ej. generados por nodos y sus enlaces de comunicación) mediante el uso de estructuras de *datos de conectividad local* (ej. protocolos de comunicación).

2.1. Enfoque topológico de un SDM

Un sistema distribuido contiene un vasto y complejo conjunto de posibles ejecuciones llamado *complex*, las cuales permiten modelar los datos, y para su descripción se fragmentan en piezas discretas llamadas *simplices* (*simplex* en singular). Como el cognado del inglés lo indica, un complex es un objeto complicado bajo análisis, por lo que se recurre a herramientas de la topología combinacional para solo percibir sus propiedades esenciales.

Además, se sabe que los modelos de SDM's difieren en términos de su confiabilidad y de su garantía de tiempo (de procesamiento y retrasos en la comunicación). Considerando esta idea, estos términos son a menudo reflejados como “huecos” en el simplicial complex.

2.1.1. Elementos de la topología combinacional

De la topología se derivan varias ramas, las cuales difieren en la forma de representar los espacios, así como en las transformaciones continuas llevadas a cabo para preservar las propiedades esenciales. La *topología combinacional* es una de las ramas, y comprende los espacios hechos de piezas sencillas o simplices, para los cuales sus propiedades esenciales pueden ser caracterizadas mediante un conteo (como la suma de los grados en los nodos de un grafo).

Un *simplicial complex* puede ser visto como una realización de mayor dimensión de un grafo. Además, los *simplicial complexes* pueden ser representados dentro del espacio Euclidiano, debido a que los vértices son representados como puntos coordenados específicos, y los *simplices* como su envolvente convexa. Así, un 0 -*simplex* es un vértice (nodo para un SDM), un 1 -*simplex* es un borde entre dos vértices, un 2 -*simplex* es un triángulo formado por 3 bordes y 3 nodos; y *simplices* de mayor dimensión son definidos de forma análoga. Estas construcciones son útiles para el modelado de la geometría subyacente a una red de sensores [8, pag. 33].

Los simplicial complex abstractos como elementos topológicos base en las redes

Definición 2.1.1. Dado un conjunto finito \mathcal{N} de nodos y una familia X de subconjuntos finitos de \mathcal{N} , se dice que X es un **simplicial complex abstracto** en \mathcal{N} si: por cada $\sigma \in X$ y $\tau \subset \sigma$, se tiene que $\tau \in X$.

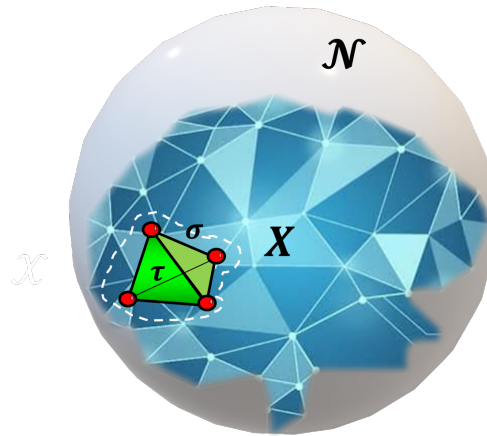


Figura 2.1: Elementos del simplicial complex abstracto por la definición 2.1.1

De la definición 2.1.1, los elementos $\sigma \in X$ son llamados **simplices**, cuya dimensión es igual a su cardinalidad menos 1: $\dim(\sigma) = |\sigma| - 1$. Se le llama **face** de un simplex σ en un simplicial complex X al simplex $\tau \subset \sigma$ de dimensión $\dim(\tau) = \dim(\sigma) - 1$. Para τ como un *face* de σ se dice que σ es un **coface** de τ .

Por otro lado, cada simplicial complex tiene una *realización geométrica* denotada por $|X|$, la cual es un espacio topológico obtenido por el “pegado” de copias disjuntas de k -*simplex* comunes, Δ^k , a lo largo de sus *faces*. Otra forma de encontrar en la literatura esta realización geométrica es el denominado **cell complex** o *d-celda*, el cual es un poliedro convexo homeomorfo a un disco de dimensión d , y es posible ensamblar celdas para construir una celda compleja (complex). En adelante, para la descripción topológica combinatorial de una celda compleja, se usará una forma más estructurada llamada *simplicial complex*, cuyas celdas consisten de vértices, bordes, triángulos, y sus extensiones de mayor dimensión. Estas construcciones son útiles para el modelado de la geometría subyacente a una red de sensores [8, pag. 33].

Para mayor detalle de los conceptos derivados de la definición 2.1.1, se puede consultar el apéndice A.

2.2. Modelos de sistemas con memoria compartida

2.2.1. Los snapshots inmediatos como interacción para la resolución de tareas

En modernos procesadores multinúcleo, la memoria es una secuencia de paquetes de bits llamadas palabras, las cuales son individualmente direccionables. Sin embargo, en vez de considerar la lectura de palabras individuales, se asume la capacidad de leer una secuencia larga y arbitraria de palabras contiguas en un solo paso atómico. Esta operación recibe el nombre de **snapshot**.

Así, se combinan las operaciones de escritura y snapshot en una operación llamada **snapshot inmediato**, en la cual cada paso puede ser de forma concurrente con otros procesos, y es como se describe a continuación:

1. Un proceso escribe su vista local a una palabra en memoria.
2. Se lleva a cabo un snapshot en toda o parte de la memoria.

Sin embargo, este modelo basado en snapshots es considerado poco realista, ya que los multiprocesadores modernos no proporcionan snapshots directamente; pero se procede a su aplicación por las siguientes razones:

1. Los snapshots inmediatos simplifican fronteras inferiores¹, debido a que estos dan lugar a estructuras combinatoriales más simples que operaciones de lectura-escritura de palabras individuales.
2. Estas operaciones no afectan la forma de solución a las tareas (si las tareas tienen solución con modelos basados en snapshots, también tendrán solución mediante instrucciones de escritura por palabra simple).

Una **tarea** es un problema a ser distribuido entre varios agentes (nodos, procesos), en el cual cada proceso inicia con un valor privado de entrada, con la posibilidad de comunicación con otro proceso, y entonces cada proceso salta con un valor privado de salida. A partir de este punto, se restringe la atención en tareas llamadas *colorless task* (tareas incoloras en adelante), en las cuales solo importa qué conjuntos de valores de entrada se asignaron y qué conjuntos de valores de salida se eligieron. Este tipo de tareas cubren muchos de los problemas centrales en computación distribuida, como el consenso o el k-acuerdo establecido.

Además, un **protocolo** es un programa que resuelve una tarea. Para los modelos descritos en este apartado, se pone interés en los protocolos *wait-free* (sin esperar), donde cada proceso debe completar su comunicación en un número limitado de pasos, y no pueden esperar por algún otro proceso.

2.2.2. Forma operacional de sistemas con memoria compartida

Los procesos

[14, Cap. 3] Se parte de que un *proceso* es una entidad computacional secuencial modelada como una máquina de estados. Así, el conjunto de $n + 1$ *procesos* comunicados en un medio (memoria compartida o paso de mensajes en red) recibe el nombre de *sistema distribuido*, donde cada uno de los procesos constan de los siguientes elementos:

¹Si una tarea es imposible de resolver utilizando snapshots inmediatos, es también imposible de resolver utilizando lecturas-escrituras de registros individuales

- Un único nombre P tomado de un universo Π de nombres: $P \in \Pi$. Para el caso simple, Π es solamente $[n] = \{0, 1, \dots, n\}$. Incluso cuando $|\Pi| \gg n + 1$, se le refiere también al proceso con un nombre i (el i -ésimo proceso), el cual es un autómata con un conjunto de estados Q_i (incluye estados iniciales Q_i^{in} y finales Q_i^{fin}). No es necesario que Q_i sea finito, ya que los procesos pueden iniciar valores de entrada de un conjunto contable, y estos pueden cambiar su estado sobre un posible número infinito de ejecuciones.

Al haber comunicación, cada proceso envía su nombre propio, por lo que todos los procesos pueden conocer los nombres unos de otros.

- Cada estado q del proceso contiene:
 - Un **nombre** inmutable denotado por $name(q) = P$, el cual toma su valor de Π . Si en una ejecución el proceso va desde el estado q hasta el estado q' : $name(q) = name(q')$.
 - Un componente **view** denotado por $view(q) = v$, el cual normalmente cambia de estado en estado sobre una ejecución, y representa que es lo que el proceso conoce acerca de la computación (por ejemplo, saber qué variables locales son usadas).

Configuraciones y ejecuciones: transiciones estructuradas de estado

Una *configuración* C es un conjunto de estados del proceso, correspondientes a la condición del sistema en un momento determinado. Cada proceso aparece a lo sumo una vez en C .

Por ejemplo, si $\{s_0, s_1\}$ son dos estados diferentes en C_i , entonces $name(s_0) \neq name(s_1)$. Por consiguiente, cada componente de conjunto $names(C)$ es inmutable en cada configuración. Además, se tiene un subconjunto $active(C)$, compuesto por los estados diferentes al estado final. A veces, en la configuración se incluye un *entorno* que indica el tipo de canales de comunicación, como el estado de la memoria compartida o los mensajes en la red.

Por otro lado, una *ejecución* define el orden en que se comunican los procesos, y corresponde a una secuencia alterna de configuraciones y conjuntos de nombres de procesos:

$$C_0, S_0, C_1, S_1, \dots, S_i, C_{i+1} \tag{2.1}$$

, donde:

- $C_0 \triangleq$ Configuración inicial.
- $S_i \triangleq$ Conjunto de nombres de procesos cuyos estados cambian entre C_i y C_{i+1} .

La secuencia S_0, S_1, \dots, S_i que genera la ejecución recibe en nombre de *itinerario*, y a la terna C_i, S_i, C_{i+1} se le nombra *paso actual*. Si $P \in S_i$, se dice que P toma un paso. En este apartado se le refiere al paso P como un snapshot inmediato.

Los procesos que aparecen en un paso se dice que son *participantes* del mismo, y solo estos procesos pueden cambiar su estado como resultado de su comunicación. Además, si la última ejecución de la configuración no es la determinada como final (debido a que esta incluye procesos cuyos estados no son los finales), entonces se considera que esos procesos han fallado.

Las tareas incoloras

Una *tarea* es un problema a ser distribuido entre varios procesos, en el cual cada proceso inicia con una asignación de entrada $A = \{(P_j, v_j) \mid P_j \in \Pi, v_j \in V^{in}\}$. Además, puede haber comunicación con otros procesos, por lo que cada proceso cambia con un valor privado de salida.

Sin embargo, se restringe la atención en tareas llamadas *incoloras*, en las cuales importa qué conjuntos de valores de entrada se asignaron (tomados de un dominio $V^{in} : \sigma = \{v_j \mid (P_j, v_j) \in A\}$) y qué conjuntos de valores de salida se eligieron (tomados de un dominio V^{out}). Este tipo de tareas cubren muchos de los problemas centrales en computación distribuida.

De manera informal se define así una tarea incolora:

Definición 2.2.1. Una *tarea incolora* es una terna $(\mathcal{I}, \mathcal{O}, \Delta)$, donde:

- $\mathcal{I} \triangleq$ Conjunto de asignaciones colorless de entrada.
- $\mathcal{O} \triangleq$ Conjunto de asignaciones colorless de salida.
- $\Delta : \mathcal{I} \mapsto 2^{\mathcal{O}}$.

Protocolos para tareas incoloras

En un *protocolo colorless* (*protocolo incoloro* en adelante) cada proceso realiza un seguimiento del conjunto de vistas que recibió, sin considerar que procesos enviaron tales vistas, y su cómputo está dividido en dos partes:

1. *Protocolos de información completa con tarea independiente.*- Aquí cada proceso envía y recibe reiteradamente vistas a otros procesos; además de actualizar su propio estado y reflejar que es lo que ha aprendido.
2. *Mapa de decisión de tarea dependiente.*- Cuando se han producido suficientes capas de comunicación, por este mapa cada proceso elige un valor de salida para su vista final.

Como se muestra en el algoritmo 1, cada proceso ejecuta un *protocolo incoloro* por etapas (en adelante *capas*), para el cual los procesos comparten un arreglo bidimensional en memoria $mem[l][i]$, donde las filas corresponden a las capas del protocolo y las columnas a los procesos. En la capa l , el proceso P_i toma un snapshot inmediato: se escribe en $mem[l][i]$, e inmediatamente se toma un snapshot de la fila l . Además, estos protocolos son de *comunicación cerrada*, en el sentido de que la información fluye desde capas anteriores hacia capas posteriores, pero no en la dirección opuesta. Así, se tiene la siguiente definición informal:

Definición 2.2.2. Un *protocolo incoloro* corresponde a la terna $(\mathcal{I}, \mathcal{P}, \Theta)$, donde \mathcal{I} es un conjunto de asignaciones colorless de entrada, \mathcal{P} es el conjunto de configuraciones colorless finales alcanzables desde \mathcal{I} , y $\Theta(\sigma)$ es el mapa que lleva cada asignación de entrada colorless $\sigma \in \mathcal{I}$ hasta el conjunto \mathcal{P} .

Algoritmo 1 Protocolo para resolver tareas colorless con memoria compartida

- 1: **compartir** el arreglo $mem = \{[N - 1], [n]\}$ de **Value**
 - 2: $view := input$
 - 3: **para** $l := 0$ a $N - 1$ **hacer**
 - 4: **immediate**
 - 5: — $mem[l][i] := view$
 - 6: — $snap := snapshot(mem[l][*])$
 - 7: $view :=$ conjunto de valores en $snap$
 - 8: **fin para**
 - 9: **devolver** $\delta(view)$
-

Relación entre protocolos incoloros y tareas incoloras

Los procesos eligen sus valores de salida para un protocolo $(\mathcal{I}, \mathcal{P}, \Theta)$ mediante un *mapa de decisión* δ que lleva a cada vista final v de proceso hacia un valor de salida $u \in V^{out}$: $\delta(v) = u$. Este mapa se extiende desde las vistas finales hasta las configuraciones finales (conjunto de vistas finales). Así, se tienen la siguiente definición:

Definición 2.2.3. Un *protocolo incoloro* $(\mathcal{I}, \mathcal{P}, \Theta)$ con un mapa de decisión $\delta : \mathcal{P} \mapsto \mathcal{O}$ permite resolver una *tarea incolora* $(\mathcal{I}, \mathcal{P}, \Delta)$ si, para cada $\sigma \in \mathcal{I}$ y cada $\tau \in \mathcal{P}$ alcanzable desde σ (es decir, tal que $\tau \in \Theta(\sigma)$), $\delta(\tau)$ es una *asignación colorless de salida* $O \in \mathcal{O}$ permitida por la especificación de tarea $O \in \Delta(\tau)$.

Las configuraciones colorless iniciales, así como las asignaciones de entrada colorless son a menudo conjuntos de valores de entrada, por lo que se usa la misma notación \mathcal{I} para un conjunto de configuraciones iniciales de un protocolo incoloro y para un conjunto de asignaciones de entrada para tareas. En contraste, $\mathcal{P} \neq \mathcal{O}$, ya que estos están relacionados por el mapa de decisión $\delta : \mathcal{P} \mapsto \mathcal{O}$.

2.2.3. Modelo combinacional

La redefinición hacia un modelo combinacional de los elementos presentados en el modelo operacional obedece a poder trabajar con términos estáticos, los cuales guardan el comportamiento de la computación concurrente del modelo operacional, y por tanto, son más fáciles de analizar.

[14] De forma particular, la redefinición de protocolos y tareas en este modelo combinacional hace más fácil la prueba de cierto tipo de propiedades. Como ejemplo, para el análisis de protocolos incoloros, se tienen dos tipos de protocolos base: protocolos para acuerdo baricentros y protocolos para acuerdo k -conjunto. El razonamiento para estos protocolos implica el cuestionar cuales son las condiciones para su existencia.

Redefinición de las tareas incoloras

Definición 2.2.4. Una *tarea incolora* es una terna $(\mathcal{I}, \mathcal{P}, \Delta)$, donde:

- $\mathcal{I} \triangleq$ Complex de entrada, con cada simplex como subconjunto de V^{in} .
- $\mathcal{O} \triangleq$ Complex de salida, con cada simplex como subconjunto de V^{out} .

- $\Delta : \mathcal{I} \mapsto 2^{\mathcal{O}} \triangleq$ Mapa portador que satisface la condición monotonicidad:
 $\sigma \subseteq \tau \implies \Delta(\sigma) \subseteq \Delta(\tau)$.

Es decir, un proceso inicia en vértices de un simplex $\sigma \in \mathcal{I}$, y estos saltan hacia los vértices de un simplex $\tau \in \Delta(\sigma)$.

Informalmente, la *cerradura bajo contención de un simplicial complex* \mathcal{I} (si $\sigma \in \mathcal{I}$ y $\sigma' \subset \sigma$, entonces $\sigma' \in \mathcal{I}$) implica que es posible para procesos sin fallas elegir sus valores de salida incluso si otros procesos fallan antes de tomar algunos pasos. Es decir: si los procesos con entradas de σ participan en la ejecución, entonces los procesos restantes con entradas en $\sigma \setminus \sigma'$ pueden fallar antes de tomar cualquier paso, y los procesos restantes arrancarían como si la configuración colorless inicial fuera σ' . Así, con un razonamiento similar, \mathcal{O} debe ser también cerrado bajo contención.

Además, por la monotonicidad de Δ , si $\sigma' \subseteq \sigma$, entonces $\Delta(\sigma') \subseteq \Delta(\sigma)$. Así:
 $(\sigma \cap \sigma') \subseteq (\sigma \wedge \sigma')$, $\Delta(\sigma \cap \sigma') \subseteq \Delta(\sigma) \wedge \Delta(\sigma') \therefore \Delta(\sigma \cap \sigma') = \Delta(\sigma) \cap \Delta(\sigma')$.

Algunos ejemplos de tareas incoloras son:

- **Consenso.**- Aquí, cada proceso inicia con $c \in \mathbb{Z}$ valores de entrada, y todos los procesos deben acordar un solo valor de salida común. Es decir, el complex de entrada \mathcal{O} consiste en un $(c-1)$ -simplex y sus *faces* conectados; y el complex de salida \mathcal{O} es el conjunto de c vértices disjuntos.
- **Acuerdo establecido (k -acuerdo establecido)** $(\mathcal{I}, \text{skel}^{k-1}\mathcal{I}, \text{skel}^{k-1})$.-Cada valor salida de proceso debe corresponder a algún valor de entrada; pero no más de k distintos valores de salida deben ser elegidos. Esta tarea tiene un protocolo trivial si $k \geq n$.

Sea $c \in \mathbb{Z}$ el número de valores de entrada:

- El complex de entrada \mathcal{I} puede ser arbitrario, y está conformado por un solo $(c-1)$ -simplex σ y sus *faces*.
- El complex de salida \mathcal{O} está conformado por el $(k-1)$ -skeleton de σ .
- El mapa portador lleva a cada simplex de entrada σ hacia el $\text{skel}^{k-1}\sigma$.
- **Acuerdo baricéntrico iterativo** $(\mathcal{I}, \text{Bary}^N\mathcal{I}, \text{Bary}^N)$.- En esta tarea los procesos inician sobre un simplex $\sigma \in \mathcal{I}$, y ellos deciden sobre los vértices de algún simplex en la subdivisión baricéntrica $\text{Bary } \sigma$ (la cual es tratada como mapa portador), para alguna $N \in \mathbb{Z}^+$.
- Convergencia de robots.

Redefinición de los protocolos incoloros

Definición 2.2.5. Una *protocolo incoloro* es una terna $(\mathcal{I}, \mathcal{P}, \Theta)$, donde:

- $\mathcal{I} \triangleq$ Simplicial complex de entrada, donde cada simplex es una asignación colorless de entrada.
- $\mathcal{P} \triangleq$ Simplicial complex de protocolo, donde cada simplex es una configuración colorless final.
- $\Theta : \mathcal{I} \mapsto 2^{\mathcal{P}} \triangleq$ Mapa de ejecución, tal que $\mathcal{P} = \cup_{\sigma \in \mathcal{I}} \Theta(\sigma)$.

Θ es un mapa portador estricto, ya que: $\Theta(\sigma \cap \sigma') = \Theta(\sigma) \cap \Theta(\sigma')$. Esto es:

1. $\Theta(\sigma) \cap \Theta(\sigma')$ es un conjunto de configuraciones colorless finales en las que ningún proceso puede decir si la ejecución comenzó con entradas de σ o de σ' . Así, solo los procesos con entradas de $\sigma \cap \sigma'$ pueden participar, debido a que los otros conocen cual fue la configuración inicial.

2. Pero las ejecuciones en 1 son las mismas que con las configuraciones finales $\Theta(\sigma \cap \sigma')$, correspondientes a las ejecuciones en las cuales solo los procesos con entradas de $\sigma \cap \sigma'$ pueden participar.

Redefinición de la relación entre protocolos incoloros y tareas incoloras

Definición 2.2.6. Un *protocolo incoloro* $(\mathcal{I}, \mathcal{P}, \Theta)$ permite resolver una *tarea incolora* $(\mathcal{I}, \mathcal{P}, \Delta)$ si hay un mapa simplicial $\delta : \mathcal{P} \mapsto \mathcal{O}$, tal que $\delta \circ \Theta$ es conllevado por Δ .

Cada simplex en \mathcal{P} es una configuración colorless final (el conjunto de estados finales que pueden ser alcanzados en alguna ejecución). Las asignaciones colorless de salida de las tareas son los simplices de \mathcal{O} .

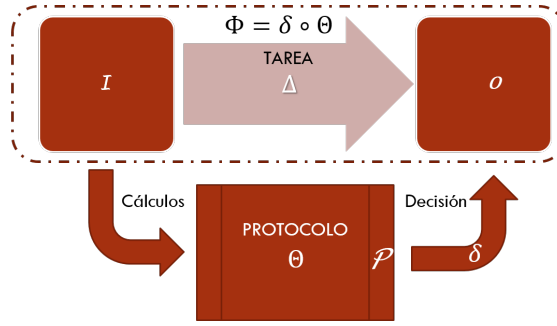


Figura 2.2: Esquema de la relación entre protocolos y tareas.

Composición de protocolos

Definición 2.2.7. (Composición de dos protocolos) Sean dos protocolos: $(\mathcal{I}, \mathcal{P}, \Theta)$ y $(\mathcal{I}', \mathcal{P}', \Theta')$, con $\mathcal{P} \subseteq \mathcal{I}'$. De su composición resulta el protocolo $(\mathcal{I}, \mathcal{P}'', \Theta'')$, donde $\Theta'' = \Theta \circ \Theta'$, esto es, $(\Theta' \circ \Theta)(\sigma) = \Theta'(\Theta(\sigma))$ para $\sigma \in \mathcal{I}$; y $\mathcal{P}'' = \Theta''(\mathcal{I})$.

Informalmente, en la definición 2.2.7 los procesos participan en el primer protocolo, y después participan en el segundo, mediante el uso de sus vistas finales del primer protocolo como sus entradas para el segundo protocolo.

Definición 2.2.8. (Composición de un protocolo y una tarea) Sea un protocolo $(\mathcal{I}, \mathcal{P}, \Theta)$ y una tarea $(\mathcal{P}', \mathcal{O}, \Delta)$, con $\mathcal{P} \subseteq \mathcal{P}'$ y Δ estricto. De su composición resulta el protocolo $(\mathcal{I}, \mathcal{O}', \Delta \circ \Theta)$, donde $(\Delta \circ \Theta)(\sigma) = \Delta(\Theta(\sigma))$, para $\sigma \in \mathcal{I}$; y $\mathcal{O}' = (\Delta \circ \Theta)(\mathcal{I})$.

Informalmente, en la definición 2.2.8 los procesos que participan en el primer protocolo usan sus vértices de salida como entradas hacia algún protocolo que resuelve una tarea.

Definición 2.2.9. (Composición de una tarea y un protocolo) Sea la tarea $(\mathcal{I}, \mathcal{O}, \Delta)$, con Δ estricto, y un protocolo $(\mathcal{I}', \mathcal{P}, \Theta)$, con $\mathcal{O} \subseteq \mathcal{I}'$. De su composición resulta el protocolo $(\mathcal{I}, \mathcal{P}', \Theta \circ \Delta)$, donde $(\Theta \circ \Delta)(\sigma) = \Theta(\Delta(\sigma))$, para $\sigma \in \mathcal{I}$; y $\mathcal{P}' = (\Theta \circ \Delta)(\mathcal{I})$.

Informalmente, en la definición 2.2.9 los procesos que participan en algún protocolo que resuelve alguna tarea, usan sus vértices de salida como entradas hacia el segundo protocolo.

Lema de protocolo complex

El siguiente lema muestra la conexión entre las estructuras discretas y continuas de tareas y protocolos. Este lema es válido para cualquier modelo computacional con protocolos que resuelven el acuerdo baricéntrico.

Lema 2.2.10. *Considerar que para cualquier complex de entrada \mathcal{I} y cualquier $N > 0$, existe un protocolo que resuelve la tarea del acuerdo baricéntrico. $(\mathcal{I}, \text{Bary}^N \mathcal{I}, \text{Bary}^N)$. Entonces, una tarea $(\mathcal{I}, \mathcal{O}, \Delta)$ tiene un protocolo si y solo si allí existe un protocolo $(\mathcal{I}, \mathcal{P}, \Theta)$ con el siguiente mapa continuo:*

$$f : |\mathcal{P}| \mapsto |\mathcal{O}| \quad (2.2)$$

tal que $f \circ \Theta$ es llevado por Δ .

2.3. Enfoque homológico de un SDM

La homología es la forma más simple para construir invariantes topológicos ² para el análisis de alguna estructura particular de datos. En esta teoría, un simplicial complex (el objeto topológico fundamental) es reemplazado con una cadena complex (el dispositivo algebraico fundamental).

[12]En su noción más simple, la homología de un espacio X construye una secuencia de espacios vectoriales $H_*(X)$, mediante descomposición de un espacio en objetos singulares (celdas o simplexes), así como su conteo y cancelación de acuerdo a la paridad de la dimensión. Las características formales de la homología, así como los elementos homológicos implementados para este trabajo, son presentados en el apéndice B.

[8]Cualquier red de comunicación sobre un conjunto de nodos \mathcal{N} tiene la estructura de un simplicial complex, con todos sus simplexes de dimensión cero o uno. Además, es posible trabajar con simplicial complexes de mayor dimensión ($\mathcal{N} \in \mathbb{R}^n$) cuando se tienen estructuras de datos complejas, como se lista a continuación:

- **Čech complex** $C_\epsilon \in \mathcal{N}$.- Simplicial complex cuyos k -simplexes son generados por $k+1$ nodos, sobre los cuales se forman bolas de diámetro ϵ , cuya intersección es mutua y no vacía.
- **Vietoris-Rips complex** $R_\epsilon \in \mathcal{N}$.- Simplicial complex cuyos k -simplexes son generados por $k+1$ nodos, sobre los cuales se forman bolas de diámetro ϵ , cuyas intersecciones son por pares y no vacías. Es decir, sus simplexes son tuplas de nodos con distancia $d \leq \epsilon$ por pares.
- **Flag complex** \mathcal{F} .- Dada una red de bordes basada en \mathcal{N} , el flag complex es el máximo simplicial complex que puede representar a \mathcal{N} como su 1-skeleton, es decir, el 1-skeleton consiste en los vértices y bordes suficientes para definir el resto de \mathcal{N} .

La riqueza de los métodos homológicos aplicados radica en configuraciones de SDM's donde se carece o fallan los métodos de localización (GPS o videocámaras, por ejemplo); además de mostrar la forma en que algún conjunto de datos o subsistemas se comportan.

²Se trata de una propiedad de los espacios topológicos que son invariantes sobre homeomorfismos.

2.3.1. Problemas de cobertura

Se asume que cada nodo n_α percibe y cubre un vecindario \mathcal{U}_α dentro de un dominio D .

- **Cobertura general.**- Se resuelve el problema de saber si la unión de todos los vecindarios contienen o son equivalentes al dominio de interés: $\bigcup_\alpha \mathcal{U}_\alpha \approx D$.
- **Cobertura obstáculo.**- Se resuelve el problema de saber si la unión de todos los vecindarios forman un obstáculo entre dos regiones de D para aislar alguna de ellas.
- **Cobertura por barrido.**- Similar a la cobertura general, pero en este caso se considera que los nodos cambian de posición en determinados tiempos: $\bigcup_{\alpha,t} \mathcal{U}(t)_\alpha \approx D$.
- **Cobertura óptima.**- Se permite calcular el número óptimo de nodos para cubrir D .

2.3.2. Criterio homológico fundamental de cobertura

En este estudio inicial se considera al grafo de conectividad $\mathcal{G}(V_\epsilon, E) \in \mathbb{R}^2$, donde V_ϵ es el conjunto de nodos (cada uno con radio de cobertura ϵ) y E corresponde al conjunto de bordes que representan la unión de dos nodos separados a una distancia ϵ . Si se conocen todas las regiones de cobertura de los nodos \mathcal{U}_β con sus intersecciones, se puede usar la homología Čech y el lema de Nerve para la búsqueda de huecos, y así dar solución al problema de cobertura general. Sin embargo, tales datos no son fácilmente conocidos [11, 2010].

En [8, pags. 33-34] Ghrist da las siguientes consideraciones poco restrictivas acerca de los enlaces de cobertura para comunicación entre nodos:

- “Cualquier terna de nodos en comunicación por pares tiene su envolvente convexa en \mathbb{R}^2 contenida en la región de cobertura”, como se muestra en la figura 2.3. Estas regiones se asumen correlacionadas a las distancias de comunicación. Este es un modelo simple, pero motiva el

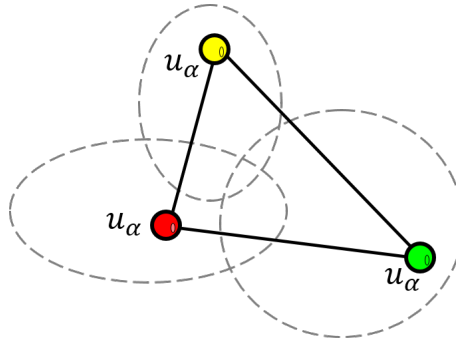


Figura 2.3: Realización geométrica de la región de cobertura de una terna de nodos en \mathbb{R}^2 .

uso del flag complex \mathcal{F} del grafo de comunicación \mathcal{G} para modelar la topología de la región de cobertura.

- Un ciclo \mathcal{C} en la red es elegido tal que su imagen en \mathbb{R}^2 sea una curva cerrada simple que acote un dominio poligonal \mathcal{D} . Así, \mathcal{C} actúa como un cercado para el problema de cobertura.

Con base en estas consideraciones, el siguiente teorema da un criterio fundamental para cobertura, basado en la homología relativa de \mathcal{F}, \mathcal{C} :

Teorema 2.3.1. *Dada una red plana de sensores caracterizada por \mathcal{G}, \mathcal{C} y \mathcal{D} (ya descritos), entonces toda la región \mathcal{D} está completamente cubierta por los sensores si:*

1. $[\mathcal{C}] = 0 \in H_1(\mathcal{F})$.
2. Existe un ciclo explícito relativo $[\zeta] \in H_2(\mathcal{F}, \mathcal{C})$, con $\partial\zeta = \mathcal{C}$.

Observación 2.3.2. El segundo criterio del teorema 2.3.1 es más aplicable en la práctica, debido a que si $\zeta \in Z_2(\mathcal{F}, \mathcal{C})$ es calculado, este provee garantía de cobertura incluso cuando los nodos no implicados en ζ son removidos.

2.4. Comportamiento dinámico en tiempo real

Las estrategias de tiempo real están basadas en tener sistemas con respuesta acotada durante su operación dinámica. Un SISTEMA DE TIEMPO REAL (RTS) la interacción entre sus elementos es con una certeza de respuesta temporal (tanto local como global) a través de los elementos de procesamiento (procesos) [3]. Así, los sistemas distribuidos cuya funcionalidad está restringida por tiempos son conocidos como SISTEMAS DISTRIBUIDOS EN TIEMPO REAL.

Una APLICACIÓN EN TIEMPO REAL está caracterizada por un conjunto Γ de n tareas RTT asociadas $\{\tau_0, \tau_1, \dots, \tau_n\}$ que deben ser completadas para lograr un comportamiento deseado. Por esto, una tarea está compuesta por una secuencia de activaciones idénticas a_i ($i = 1, \dots, n$) separadas entre ellas por un lapso de tiempo.

Así, los parámetros que una RTT son:

- **Tiempo de llegada o de liberación (r_i).**- Tiempo en que la tarea es establecida para su ejecución.
- **Tiempo de consumo (c_i).**- Tiempo que le toma a cada tarea para ejecutar sus actividades asignadas sin interrupción.
- **Plazo de entrega (*deadline*) (d_i).**- Límite de tiempo en que la tarea debe de finalizar. De este parámetro, se tienen las siguientes definiciones:
 - TAREA PROGRAMABLE.- Tarea que completa su objetivo en un tiempo menor o igual a d_i .
 - SISTEMA PROGRAMABLE.- RTS en que todas sus tareas terminan antes de sus deadlines.
- **Tiempo de inicio (s_i).**- Tiempo en que la tarea inicia su ejecución.
- **Tiempo de finalización (f_i).**- Tiempo en que la tarea termina su ejecución. Así, una tarea es programable si $f_i \leq d_i$.
- **DESPLAZAMIENTO (φ_i).**- Tiempo de la primera instancia periódica producido cuando la tarea tiene un retraso por el cual no inicie exactamente en el tiempo s_i .

Las TAREAS EN TIEMPO REAL (RTT) son entidades ejecutables de trabajo J_i que al menos son caracterizadas por un tiempo de ejecución máximo y una restricción de tiempo, y está formada por un conjunto de instancias $J_i = \{J_{i,k}\}$, con $i, k \in \mathbb{Z}^+$.

Una INSTANCIA $\{J_{i,k}\}$ es una unidad de trabajo de una tarea, y está definida por la siguiente tupla:

$$J_{i,k} = (r_{i,k}, c_{i,k}, d_{i,k}), i, k \in \mathbb{Z}^+ \quad (2.3)$$

Cuando solo se desarrollará una sola tarea, se puede prescindir del índice i .

Capítulo 3

ANÁLISIS PARA EL MODELO DESARROLLADO

Ya con los fundamentos de topología combinatorial, y considerando el trabajo relacionado al modelado de SDM's, en este capítulo se detalla el análisis realizado para el modelado a realizar, considerando restricciones de tiempo real.

3.1. Representación general del SDM

Se considera el caso en que los nodos viven en un dominio euclidiano plano $D \subset \mathbb{R}^2$ compacto y con frontera poligonal. Cada nodo puede comunicarse dentro de cierta vecindad simétrica y radial, para así realizar determinada tarea. Para resolver el problema sobre si la unión de todos los discos alrededor de los nodos cubren D , se aplican métodos topológicos, con los cuales no es necesario que los nodos cuenten con alguna forma de localización, orientación o medición de distancia.

3.1.1. Marcos y consideraciones de operación

Los siguientes marcos operación y distribución de los n nodos fueron adoptadas de Ghrist, Muhammad y Jadbabaie, ya que garantizan capacidades de cobertura y comunicación basadas en proximidad.

Cobertura simplicial

Cada nodo puede comunicarse con otros nodos dentro de un dominio simétrico radial r_b . Por lo anterior, cualquier subconjunto de nodos que estén comunicados por pares cubre su conjunto convexo Q , expresado en la ecuación 3.1:

$$\mathcal{A}(V) = \bigcup \{conv(Q) \mid Q \subseteq V, \max_{v_i, v_j \in Q} \|v_i - v_j\|_2 \leq r_b\} \quad (3.1)$$

donde V es el conjunto de nodos, y v_i la localización estimada de i -ésimo nodo.

Cobertura simétrica

- Similar a marco de cobertura simplicial, pero ahora asumiendo que cada nodo cubre áreas simétricas radiales r_c , expresadas en la ecuación 3.2:

$$\mathcal{U}(V) = \bigcup_{v \in V} \{x \in \mathbb{R}^2 : \|x - v\| \leq r_c\} = \bigcup_{v \in V} U_v \quad (3.2)$$

donde U_v es el disco de cobertura del nodo localizado en el punto v .

- La región de interés \mathcal{D} de cada cúmulo es conectada y compacta, y tendrá cobertura completa si es un subconjunto de $\mathcal{U}(V)$.
- Cada \mathcal{D} estará acotada linealmente por una frontera $\partial\mathcal{D}$, donde se ubicarán sensores denominados *fence*. Cada uno de estos nodos identificará a sus dos nodos *fence* vecinos, por lo que se formará una configuración cíclica bien definida.
- Por el trabajo de Ghrist, se asume que $r_b \leq r_c\sqrt{3}$.

Consideraciones generales de operación

Como se observa en la figura 3.1, se consideran los siguientes elementos dentro del espacio donde se desarrolla la red de nodos móviles:

- $n_{LETRA'_n}$.- Id del nodo, donde “LETRA” denota el cúmulo al cual pertenece, y $n \in \mathbb{Z}$ es el número de nodo.
- r_c .- Radio de cobertura del nodo.
- $r_{c'LETRA'}$.- Radio de cobertura del nodo “LETRA”.

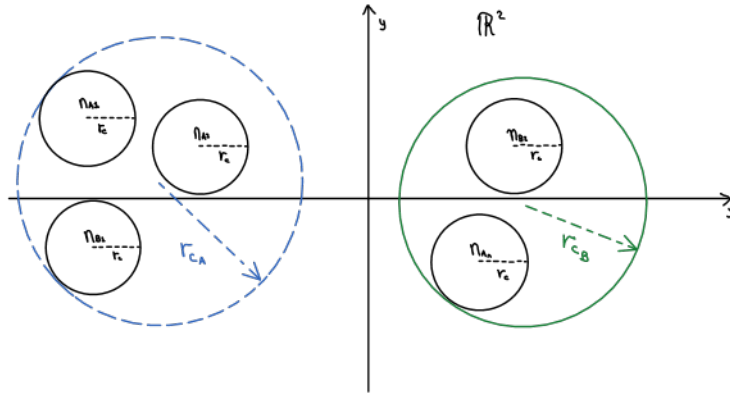


Figura 3.1: Esquema en \mathbb{R}^2 de los nodos a considerar en este análisis

Consideraciones a tomar en cuenta:

- Cada nodo tiene asignado un único *ID* a ser transmitido.
- Cada nodo es capaz de comunicar su *ID* a sus nodos vecinos.
- Cada nodo tiene un *ID* dentro de un arreglo, el cual corresponde al simplex del cuál el nodo es parte.
- Se tienen rangos radiales limitados de cobertura por nodo (r_c) y por cúmulo ($r_{c'LETRA'}$).
- Sobre la forma de detección entre nodos, esta posible ya sea por la detección de una señal fuerte en un radio r_s , o por la detección de una señal débil en un radio r_w .
- Los vértices de la frontera poligonal $\partial\mathcal{D}$ son definidos por nodos especiales llamados *nodos fence*, los cuales forman una configuración cíclica bien definida. Además, cada uno de estos nodos identifica sus dos *nodos fence* vecinos.

Ya con la premisas consideradas, los nodos de cada cúmulo son representados por un grafo dirigido G_{LETRA} , cuyos vértices corresponderán a los nodos de la red móvil, y los bordes serán la distancia estimadas de separación, de acuerdo a los radios de detección r_s y r_w .

Ahora bien, para obtener la caracterización topológica que reemplace la definición operacional por grafos, se debe probar la existencia de un mapa continuo entre espacios topológicos, mediante la captura de los conjuntos de posibles entradas y salidas de la tarea distribuida a realizar [9]. Para esto, se cambia a una representación por simplicial complex abstracto, cuyos simplices son determinados al rellenar los grafos correspondientes: cada uno será el simplicial complex mayor con el correspondiente grafo como su $1 - d$ skeleton [7].

$$G_{LETRA} \mapsto \mathcal{G}_{LETRA} \quad (3.3)$$

La representación topológica es válida, ya que un simplicial complex es una generalización de un grafo, ya que se captura relaciones de orden superior entre los elementos de V . De acuerdo con de Silva y Ghrist [7], una forma de representación de grafos dirigidos en simplicial complexes es por medio de los *Vietoris-Rips complex* (o como mejor se les conoce, *Rips-complex* \mathcal{R}), los cuales pueden ser definidos para cualquier espacio métrico \mathbf{M} y cualquier distancia δ , por la formación de un simplex para cada conjunto finito de puntos que tengan un diámetro de al menos δ . Es decir, se trata de una familia de subconjuntos finitos de \mathbf{M} , siendo cada uno un subconjunto de k puntos como parte de un $(k - 1) - dimensional$ simplex.

3.1.2. Cómputo descentralizado para la estimación de invariantes topológicos

[21] La computación de ciertos invariantes topológicos en redes de sensores es relevante para el modelado de la cobertura y el ruteo.

[21] La idea de un algoritmo distribuido para el cómputo de invariantes globales podría parecer contradictoria, ya que si el tamaño de los huecos de comunicación es relativamente largo, se tendría la necesidad de acumular la información local en una cierta región antes de del cómputo del conocimiento global de la topología. Por lo tanto, el o los nodos que porten toda la información local podrían ejecutar algún algoritmo homológico centralizado y calcular la topología global. Sin embargo, para el caso de las redes de sensores, donde sus agentes tienen una limitada potencia computacional, la acumulación de datos de red en uno o pocos nodos no es viable, debido a las limitaciones de los recursos en la red.

Como forma de resolver la problemática planteada, se toman en cuenta ciertos métodos probabilísticos, por los cuales se crean procesos aleatorios a nivel de nodo, para así ponderar las estimaciones locales, y que converjan con alta probabilidad hacia el conocimiento global del sistema. Además, para el modelado de la potencia de transmisión a ser detectada por los nodos, se toma como base la tesis en [2], en la cual se describe la técnica del control de potencia de transmisión (TPC). En esta técnica, el nodo móvil ajusta su potencia de transmisión, de forma que únicamente el nodo con el que quiere comunicarse pueda escuchar sus transmisiones. Así, su viabilidad es analizada mediante un modelo probabilístico, el cual permite a un nodo móvil estimar la potencia de transmisión que maximice la probabilidad de que su señal inalámbrica alcance el punto de acceso (AP) más cercano.

3.2. Modelo de protocolos que resuelven tareas

Se parte de que una tarea \mathcal{T} es solucionable en un modelo \mathcal{M} si existe un protocolo \mathcal{P} por el cual, en cada ejecución de \mathcal{M} , cada proceso toma suficientes pasos y eventualmente produce salidas, tal que

tales salidas cumplan con las especificaciones de la tarea con respecto a las entradas proporcionadas [9, 2014]. Se tratará al modelo como un conjunto de ejecuciones, en las cuales se intercalaran snapshots inmediatos.

Manejo de fallas En un SDM las fallas pueden estar correlacionadas para procesos que se ejecutan en el mismo nodo, que se ejecutan en la misma partición de red o que son administrados por el mismo proveedor. Para un modelo básico, el objetivo es proporcionar algoritmos sin esperas (*wait-free*) que resuelvan tareas particulares cuando cualquier número de procesos puede fallar; sin embargo, este enfoque es muy exigente y, a veces, sobrado.

Proposición 3.2.1. *Un protocolo t -resiliente es aquel que funciona correctamente cuando el número de procesos defectuosos no supera un valor t , $0 \leq t \leq n$ procesos. Un algoritmo sin esperas para $n + 1$ procesos es n -resiliente. Así, una tarea incolora tiene un protocolo t -resiliente en un modelo si, para todo $n \geq t$, hay un protocolo t -resiliente con $(n+1)$ -procesos para esa tarea.*

Proposición 3.2.2 (Lema Skeleton). *Si un protocolo $(\mathcal{I}, \mathcal{P}, \Theta)$ resuelve una tarea $(\mathcal{I}, \mathcal{O}, \Delta)$ y hay un protocolo tipo k -set agreement para \mathcal{I} , entonces la composición de k -set agreement con $(\mathcal{I}, \mathcal{P}, \Theta)$ también resuelve $(\mathcal{I}, \mathcal{O}, \Delta)$*

Observación 3.2.3. Informalmente, la proposición 3.2.2 señala que si se tiene un protocolo para una tarea, así como un protocolo para el k -set agreement, entonces se puede resolver la tarea al preprocesar con el k -set agreement.

Proposición 3.2.4 (Lema del complex protocolo). *Si hay un protocolo por capas tipo t -resiliente para $(\mathcal{I}, \mathcal{O}, \Delta)$, entonces hay un protocolo $(\mathcal{I}, \mathcal{P}, \Theta)$ tal que:*

$$\mathcal{P} = \text{Bary}^N(\text{skel}^t \mathcal{I}) \quad (3.4)$$

$$\Theta(\sigma) = \text{Bary}^N \bullet \text{skel}^t(\sigma) \quad (3.5)$$

Observación 3.2.5. Por la proposición se asume que cualquier \mathcal{P} es una subdivisión baricéntrica de \mathcal{I} .

3.2.1. Paso de mensajes entre nodos del SDM

Condiciones de operación

- Se tiene la interacción de $n + 1$ nodos/procesos, mediante el envío-recepción de mensajes en un SDM.
- Cada mensaje enviado desde un nodo p se entrega una vez a su nodo destino q mediante el procedimiento $\text{enviar}(id_p, Val, L_k, J_{i,l})$, donde:
 - id_p .- Identificador del nodo que está enviando el mensaje.
 - $val_0, \dots, val_l \in Val$.- Conjunto de valores correspondientes al desarrollo del protocolo que resuelva una determinada tarea.
 - L_k .- Conjunto de listas con los simplicial complexes de orden k a ser construidos. Este procedimiento será descrito en el subcapítulo 3.5.
 - $J_{i,l}, i, l \in \mathbb{Z}^+$.- Tupla con las restricciones de tiempo real para i ésima tarea a ser desarrollada durante el tiempo l . Esta tupla será detallada en el algoritmo 7 dentro del subcapítulo 3.3.1.

Observación 3.2.6. El procedimiento *enviar(...)* será sobrescrito, de forma que los argumentos utilizados sean de acuerdo con el protocolo a desarrollar.

- Se consideran a los procesos como *t-resilientes*, donde *t* es el número de procesos que pueden fallar.
- En cada ocasión que un proceso *p* reciba un mensaje de otro proceso *q*, *p* reenvía tal mensaje a todos los procesos (incluido el mismo): *enviar(id_q, Val, L_k, J_{i,l})* a todos.
- En el momento que un proceso *q* recibe un mensaje de un proceso *P*, este realiza una acción determinada: *al recibir (id_p, Val, L_k, J_{i,l}) hacer...*

Observación 3.2.7. El procedimiento *recibir(...)* será sobrescrito, de forma que los argumentos utilizados sean de acuerdo con el protocolo a desarrollar.

- Cada proceso debe continuar reenviando mensajes incluso después de que haya elegido su valor de salida. Sin tal garantía, un proceso no defectuoso que elige una salida y se queda en silencio no se puede distinguir de un proceso que ha fallado, lo que implica imposibilidad de realizar las tareas donde se requiera que la mayoría de los procesos no sean defectuosos. Así, se considera este reenvío continuo a ejecutarse en segundo plano, intercalado con los pasos del propio protocolo:

Algoritmo 2 :background()

- 1: *Al recibir el mensaje hacer:*
 - 2: **mientras** *recibir(id_q, Val, L_k, J_{i,l})* **hacer**
 - 3: *enviar (id_p, Val, L_k, J_{i,l})* a todos los procesos
 - 4: **fin mientras**
-

Acuerdo baricéntrico

Proposición 3.2.8. *Al tener las operaciones de envío y recepción como primitivas, se asume que cualquier protocolo complex \mathcal{P} es una subdivisión baricéntrica del complex de entrada \mathcal{I} .*

Definición 3.2.9 (Subdivisión baricéntrica). Estructura topológica combinatoria por la cual cada nodo/proceso *p* es asignado como entrada en un vértice v_p del simplex σ ; y después de intercambiar mensajes con el resto de los nodos, se elige un *face* $\sigma_p \subseteq \sigma$ que contenga a v_p , tal que para cualesquiera dos procesos participantes *p, q*, los *faces* que se elijan sean ordenados por inclusión: $(\sigma_p \subseteq \sigma_q \text{ o } \sigma_q \subseteq \sigma_p)$.

En el algoritmo 3 muestra una forma para la solución de este tipo de tarea, donde cada proceso/nodo P_i mantiene un conjunto V_i de los mensajes que ha recibido, y se desarrolla de la siguiente manera:

- Al iniciar el protocolo, solo se tienen los valores del P_i correspondiente.
- P_i continuamente transmite V_i , y espera a recibir conjuntos de otros procesos.
 - Si se recibe V' tal que $V' = V_i$, entonces se incrementa su contador del número de veces que ha recibido V_i .
 - Si se recibe V' tal que $V' \setminus V \neq \emptyset$, se asigna $V_i \cup V'$ a V_i y se reinicia el protocolo.
- Cuando P_i ha recibido $n + 1 - t$ copias idénticas de V_i provenientes de distintos procesos, el protocolo termina, y P_i decide V_i .

- Después de que el hilo principal del protocolo termina, P_i debe continuar reenviando mensajes hacia los demás procesos (ejecución en background).

Algoritmo 3 :acuerdoBaricéntrico()

Entrada: v_i - vértice a evaluar.

Salida: V_i afectado por el acuerdo baricéntrico.

```

1:  $V_i := \{v_i\}$  //Conjunto de mensajes que  $P_i$  ha recibido.
2:  $cuenta := 0$  //Contador de las confirmaciones recibidas hasta cierto momento .
3: mientras  $cuenta < n + 1 - t$  hacer
4:   //Obtener confirmación de cada proceso sin falla.
5:    $enviar(P_i, V_i)$  a todos
6:   al recibir  $(P_j, V_j)$  hacer //Recolección de respuestas.
7:   si  $V_i = V_j$  entonces
8:      $cuenta := cuenta + 1$ 
9:   si no
10:    si  $V_j \setminus V_i \neq \emptyset$  entonces
11:       $V_i := V_i \cup V_j$ 
12:       $cuenta := 0$ 
13:    fin si
14:  fin si
15: fin mientras
16: devolver  $V_i$ 
17:
18: // ejecución en background después que el protocolo devuelve  $V_i$ .
19: background
20: al recibir  $(P_j, V_j)$  hacer
21:  $V_i := V_i \cup V_j$ 
22:  $enviar(P_i, V_i)$  a todos

```

Protocolo (t+1)-set agreement (fijarAcuerdo)

Como se muestra en el algoritmo 4, al inicio cada proceso reúne valores de tantos otros procesos como sea posible. El método *getQuorum()* (desarrollado en el algoritmo 5) recopila valores hasta que recibe mensajes de todos los procesos.

Observación 3.2.10. Es seguro esperar tantos mensajes porque hay al menos $n + 1 - t$ procesos no defectuosos. No es seguro esperar más, porque los procesos t restantes pueden haberse bloqueado.

El conclusión, en el algoritmo 4 para realizar un acuerdo de conjunto (t + 1) cada proceso transmite su valor de entrada, espera recibir valores de un quórum de $n + 1 - t$ mensajes, y elige el menor valor entre ellos.

Condición de solvencia

El teorema a continuación mostrado es el principal resultado obtenido por Herlihy, Kozlov y Rajsbaum en [14], donde se muestra un criterio de existencia de un protocolo para la solución del paso de mensajes entre nodos.

Algoritmo 4 :fijarAcuerdo(v_i)

Entrada: v_i .- valor de entrada del proceso P_i .

Salida: Valor mínimo del conjunto V

- 1: enviar(P_i, v_i) a todos
 - 2: $V :=$ Conjunto de valores := *getQuorum()* // Obtener valores de todos menos t .
 - 3: **devolver** el valor mínimo de V // Es posible no haber obtenido t valores menores.
-

Algoritmo 5 :getQuorum()

- 1: $V :=$ Conjunto de valores := \emptyset
 - 2: $q : \text{int} := 0$ // Contador de los mensajes recibidos.
 - 3: **repetir**
 - 4: **hasta** recibir(Q, v) **hacer**
 - 5: $V := V \cup \{v\}$
 - 6: $q := q + 1$
 - 7: **hasta que** $q = n + 1 - t$
 - 8: **devolver** V
-

Teorema 3.2.11. [14] Para $2t < n + 1$, la tarea $(\mathcal{I}, \mathcal{O}, \Delta)$ tiene un protocolo t -resiliente si y solo si hay un mapa continuo $f : |\text{skel}^t \mathcal{I}| \mapsto |\mathcal{O}|$, el cual es llevado por Δ .

Bosquejo de la prueba.- Para mostrar que f implica \mathcal{P} , f tiene una aproximación a simplex dada por $\phi : \text{Bary}^N \text{skel}^t \mathcal{I} \mapsto \mathcal{O}$ (llevado por Δ). Por otro lado, mediante la construcción de un protocolo a dos pasos:

1. Los procesos usan el protocolo $(t+1)$ -set agreement (algoritmo 4) para converger a un simplex $\sigma \in \text{skel}^t \mathcal{I}$.
2. Los procesos repiten el barycentric agreement protocol (algoritmo 3) para converger a un simplex en $\text{Bary}^N \text{skel}^t \mathcal{I}$.

Finalmente, al realizar la composición de estos protocolos, y aplicar ϕ como un mapa de decisión δ , se produce el protocolo deseado.

3.3. Modelo de cobertura e identificación de cúmulos por simplicial complexes

Considerar el grafo $\mathcal{G}(N, E)$ de nodos móviles, cada uno con un radio de cobertura ϵ . En \mathcal{G} , N es el conjunto de vértices correspondientes a los nodos, y E es el conjunto de bordes (tal que dos nodos estén a una distancia ϵ uno del otro). Con esto, se construye el *Rip Complex* inducido por \mathcal{G} .

Definición 3.3.1. Dado un conjunto de nodos $n_0, n_1, \dots, n_k \in \mathbb{R}^n$ y $\epsilon > 0$, el *Vietoris-Rips complex* (denotado por \mathcal{R}) es el simplicial complex cuyos k -simplices corresponden a las $(k+1)$ -tuplas no ordenadas de nodos que son colocados por pares a una distancia ϵ uno del otro.

Dada la definición 3.3.1, se tienen las siguientes observaciones:

Observación 3.3.2. Los simplices de orden 0 y 1 que pertenecen a \mathcal{R} corresponden a los nodos y los vértices de \mathcal{G} , es decir, corresponde a la información acerca de la región cubierta por cada nodo.

Observación 3.3.3. La dimensión de $H_0(\mathcal{R})$ significa el número de componentes conectados de la red distribuida. Por lo anterior: $H_0(\mathcal{R})$ es de dimensión 1 si y solo si su correspondiente \mathcal{G} es conectada.

Observación 3.3.4. La dimensión de $H_1(\mathcal{R})$ significa el número de huecos en \mathcal{G} .

Cualquier subconjunto de nodos en comunicación por pares cubre todo su casco convexo, lo que implica que la región cubierta está dada por:

$$\mathcal{A}(N) = \bigcup \{conv(Q) | Q \subseteq N, \max_{n_i, n_j \in Q} \|n_i - n_j\|_2 \leq r_b\} \quad (3.6)$$

Observación 3.3.5. Por la ecuación 3.6 se concluye que el radio ϵ necesario para la construcción de \mathcal{R} es r_b , es decir: $\epsilon = r_b$.

Observación 3.3.6. Así, para la formación de cada cúmulo, interesa verificar si todos los puntos dentro de una región \mathcal{D} están cubiertos por los nodos, es decir:

$$\mathcal{D} \subseteq \mathcal{A}(N) \quad (3.7)$$

considerando además que toda la región de cobertura de los nodos está acotada por *fence nodes*:

$$\partial\mathcal{F} \subseteq \mathcal{A}(N) \quad (3.8)$$

3.3.1. Construcción de una representación local de \mathcal{R}

Se requiere conocer las condiciones para desarrollar el cálculo de los grupos homológicos de forma distribuida, por lo que se necesita un algoritmo que se ejecute a nivel de nodo e interactúe con nodos vecinos. Así, para obtener una representación local de \mathcal{R} , se tienen los siguientes dos pasos:

1. *Simplex miembro.*
2. *Simplex propiedad.*

Simplex miembro

Por el algoritmo 7 (el cual llama al algoritmo 6), cada nodo conoce de cuál simplex de mayor dimensión es parte, ya que un nodo puede pertenecer a varios simplices dentro del complex. De forma

Algoritmo 6 :encontrarSimplices()

Entrada: k ; Lista de k -simplices L_k

Salida: Lista de $k + 1$ -simplices L_{k+1}

- 1: $K :=$ longitud de L_k
 - 2: $L_{k+1} = \emptyset$
 - 3: $C :=$ todo K elige k combinaciones de $\{1, \dots, K\}$
 - 4: **para** $i := 1$ hasta $|C|$ **hacer**
 - 5: $\{\sigma_1, \dots, \sigma_k\} :=$ simplices correspondientes a la combinación $C(i)$
 - 6: $\omega :=$ la cadena $\sigma_1 + \sigma_2 \dots + \sigma_k$
 - 7: **si** $\partial_k \omega = 0$ **entonces**
 - 8: Adjuntar $\{\sigma_1, \dots, \sigma_k\}$ a L_{k+1}
 - 9: **fin si**
 - 10: **fin para**
-

general, el algoritmo 6 trabaja de la siguiente manera:

- Al inicio, cada nodo es consciente de su propio ID , por lo que se genera un conjunto de 0 – $simplices$.
- Después de esto, los nodos simultáneamente envían sus ID 's hacia los nodos vecinos dentro de su radio ϵ ; estos vecinos reciben la información y la anexan en sus respectivos arreglos de ID 's. Así, se genera el conjunto de 1 – $simplices$.
- Los nodos envían la lista de 1 – $simplices$ que poseen, y después de la recepción, cada nodo compara el 1 – $simplex$ recibido con su propio arreglo 1 – $simplex$ en búsqueda de un ciclo. Si se encuentra algún ciclo, el nodo en cuestión anexa cada ciclo a su arreglo, por lo que se construyen los 2 – $simplices$. Así, los resultados de encontrar de 0 a 2- $simplices$ son mostrados en la figura 3.2:

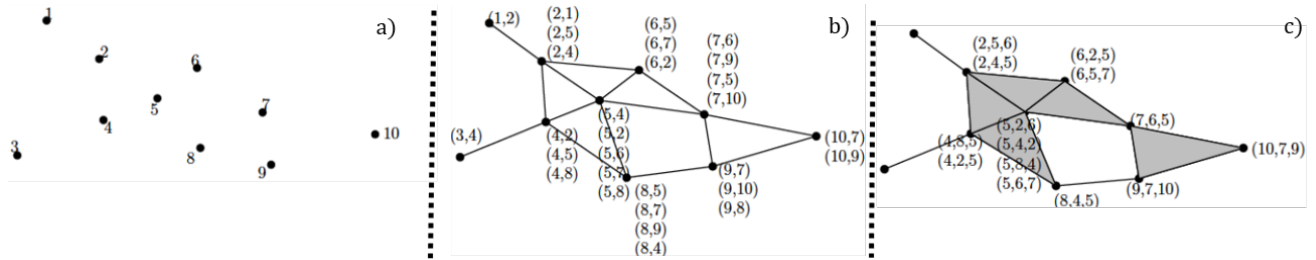


Figura 3.2: Realización geométrica del algoritmo 6 para encontrar: a) 0- $simplices$, b) 1- $simplices$, c) 2- $simplices$. Figura adaptada de [19].

- En la siguiente transmisión, la lista de 2 – $simplices$ es compartida a los nodos vecinos, para así encontrar 3 – $simplices$.
- De forma general, todos los $simplices$ de dimensión menor o igual a k son conocidos en el k -ésimo envío; por lo que desde una lista de k - $simplices$, la lista de $k + 1$ - $simplices$ es generada.

Para el cálculo de un grupo homológico n – $dimensional$, se requieren solo $simplices$ de dimensión n , $n - 1$ y $n + 1$. Así, el protocolo $simplexMiembro()$ aplica el protocolo $encontrarSimplexes()$ de forma iterativa en cada nodo, hasta que todos los $simplices$ de alguna dimensión k deseada sean encontrados. Por último, para resolver el problema de indicar a cada nodo cuando transmitir y cuándo todas las transmisiones han sido recibidas por sus nodos vecinos, un protocolo MAC es implementado.

Simplex propiedad

Ya que todos los $k + 1$ nodos pueden reclamar la pertenencia de un n - $simplex$ en particular, se requiere un protocolo de red a nivel $simplex$, como se muestra en el algoritmo 8.

Observación 3.3.7. El algoritmo 8 concede la propiedad de un $simplex$ a un nodo, si tal nodo tiene el ID menor en el $simplex$. El resultado es un subcomplex $\mathcal{R}_j \subset \mathcal{R}$ perteneciente al nodo n_j .

Algoritmo 7 :simplexMiembro()

Entrada: ID del nodo j ; la dimensión k mayor de simpleses a ser encontrados.

Salida: Lista de simpleses $\{L_1, \dots, L_k\}$

```

1:  $L_0 = \{j\}$ 
2: para  $i := 1$  a  $k$  hacer
3:    $L_i = \emptyset$ 
4:   repetir
5:     ESPERAR
6:   hasta que MAC permite TRANSMITIR
7:   TRANSMITIR( $L_{i-1}$ )
8:   repetir
9:     ESPERAR
10:  hasta que MAC indica todos los mensajes recibidos
11:  Obtener mensajes  $\mathcal{M}_{i-1} := \{M_1, \dots, M_{deg(n_j)}\}$ 
12:  Adjuntar  $L_1$  a  $\mathcal{M}_{i-1}$ 
13:   $L_i := encontrarSimpleces(\mathcal{M}_i, i)$ 
14: fin para

```

Algoritmo 8 : simplexPropiedad()

Entrada: ID del nodo j ; la dimensión mayor de simpleses para encontrar k .

Salida: $\mathcal{R}_j \subset \mathcal{R}$ perteneciente a n_j .

```

1:  $\{L_1, \dots, L_k\} \leftarrow simplexMiembro(j, k)$ ;
2:  $\mathcal{R}_j \leftarrow \emptyset$ ;
3: para  $p \leftarrow 1$  a  $k$  hacer
4:   para  $q \leftarrow 1$  a  $|L_p|$  hacer
5:      $\sigma \leftarrow L_p(q)$ ;
6:      $\{v_0, \dots, v_k\} \leftarrow$  nodos de  $\sigma$ ;
7:      $v_{min} = \min(v_0, \dots, v_k)$ ;
8:     si  $v_{min} = j$  entonces
9:       adjuntar  $\sigma$  a  $\mathcal{R}_j$ ;
10:    fin si
11:  fin para
12: fin para

```

Observación 3.3.8. El Rip complex completo resulta de la unión de todos los subcomplex, como se muestra en la ecuación 3.9:

$$\mathcal{R} = \bigcup_{j=1}^N \mathcal{R}_j \quad (3.9)$$

Resultados de la representación local de \mathcal{R}

De acuerdo con [21], por la aplicación de las propiedades especiales de los *Rip-complex*, la implementación del algoritmo 8 permite dar la siguiente proposición:

Proposición 3.3.9 (Homología trivial). *Si $\mathcal{R}_j \subset \mathcal{R}$ es un subcomplex resultado de la ejecución del algoritmo 8 sobre el nodo n_j , entonces \mathcal{R}_j tiene homología trivial.*

Prueba. Suponer que los simplices $\{\sigma_1, \sigma_2, \dots, \sigma_n\} \in \mathcal{R}_j$ *span* $C_k(\mathcal{R}_j), \forall k \geq 0$. Ya que todos los simplices tienen a j como nodo común, cualquier cadena en $C_k(\mathcal{R}_j)$ es un vecindario en forma de estrella de j . Además, si hay un ciclo inducido por el simplex en dimensión $k + 1$, por las propiedades de los Rip-Complex, siempre se rellena por la acción del propio simplex. Por lo anterior, \mathcal{R}_j está siempre conectado y $H_k(\mathcal{R}) \cong 0, \forall k > 0$. \square

Este resultado también sugiere métodos para combinar los subcomplejos en varios nodos para calcular las clases de homología global.

Proposición 3.3.10. *Si los $k + 1, k, k - 1$ simplices de \mathcal{R} están contenidos completamente en dos subcomplexes $\mathcal{R}_i, \mathcal{R}_j$ (los cuales son resultado de la ejecución del algoritmo 8 sobre los nodos i, j), entonces la k -ésima homología de $\mathcal{R} = \mathcal{R}_i \cup \mathcal{R}_j$ es isomórfica a la homología $\mathcal{R}_i \cap \mathcal{R}_j$ en una dimensión menos.*

Prueba. Al aplicar la secuencia exacta *Mayer-Veitoris* de la siguiente forma: la imagen de cada mapa

$$\begin{array}{ccccccc} \dots & \rightarrow & H_k(\mathcal{R}_i) \oplus H_k(\mathcal{R}_j) & \xrightarrow{\Psi} & H_k(\mathcal{R}) & \xrightarrow{\partial} & \dots \\ & & \dots H_{k-1}(\mathcal{R}_i \cap \mathcal{R}_j) & \xrightarrow{\Phi} & H_{k-1}(\mathcal{R}_i) \oplus H_{k-1}(\mathcal{R}_j) & \rightarrow & \end{array}$$

es isomófica al kernel de su mapa anterior. De la proposición 3.3.9 se tienen los siguientes resultados triviales: $H_k(\mathcal{R}_i) \oplus H_k(\mathcal{R}_j) \cong 0$ y $H_{k-1}(\mathcal{R}_i) \oplus H_{k-1}(\mathcal{R}_j) \cong 0$. Por exactitud, ∂ es un isomorfismo, y así $H_k(\mathcal{R}_i \cup \mathcal{R}_j) \oplus H_{k-1}(\mathcal{R}_i \cap \mathcal{R}_j) \cong 0$. \square

3.3.2. Cálculo de grupos homológicos mediante operadores Laplacianos

La descomposición espectral de operadores Laplacianos de alto orden sobre simplicial complexes fue el procedimiento seguido para el cómputo de grupos homológicos, con base en los resultados de Jadbabaie y Muhammad expresados en [21]. Estos Laplacianos pueden ser operadores locales promediados o por mezcla de operaciones, y su forma de propagación distribuida permite detectar ausencia o existencia de huecos en la cobertura del SDM. Así, se implementó un algoritmo tipo gossip para recuperar una estimación global de ciertas matrices que no puedan ser calculadas localmente debido a huecos, y así lograr calcular una descomposición espectral distribuida de los Laplacianos.

k-Laplacianos como promedio local

La idea principal es calcular el espacio nulo del primer Laplaciano combinatorio de un complejo simplicial correspondiente a la cobertura, y así extraer la información que proporciona sobre la primera homología.

Sean $\sigma_1, \sigma_2, \dots, \sigma_n \in X$ k -simplices orientados. Así, para un par de nodos i, j se obtienen los elementos $(\mathcal{L}_k)_{ij}$ expresados en la ecuación 3.10:

$$(\mathcal{L}_k)_{ij} = (\partial_k^* \partial_k)_{ij} + (\partial_{k+1} \partial_{k+1}^*)_{ij} = \left\{ \begin{array}{ll} \text{deg}_u(\sigma_i) + k + 1 & \text{si } i = j \\ 1 & \text{si } i \neq j, \sigma_i \not\sim \sigma_j \text{ y } \sigma_i \smile \sigma_j \text{ con similiar orientación} \\ -1 & \text{si } i \neq j, \sigma_i \not\sim \sigma_j \text{ y } \sigma_i \smile \sigma_j \text{ con distinta orientación} \\ 0 & \text{si } i \neq j \text{ y ya sea } \sigma_i \frown \sigma_j \text{ o } \sigma_i \not\sim \sigma_j \end{array} \right\} \quad (3.10)$$

Observación 3.3.11. El elemento $\epsilon_{ij} \in \{-1, 1\}$ captura la similitud o diferencia de orientación entre los simplices σ_i, σ_j , por lo que el cálculo del k -Laplaciano puede ser escrito a nivel simplex de la siguiente forma:

$$\mathcal{L}_k(\sigma_i) = (\text{deg}_u(\sigma_i) + k + 1)\sigma_i + \sum_{\sigma_i \smile \sigma_j} (\epsilon_{ij} \sigma_j) - \sum_{\sigma_i \frown \sigma_m} (\epsilon_{im} \sigma_m) \quad (3.11)$$

Así, la ecuación 3.11 indica que los k -Laplacianos trabajan de forma similar a un algoritmo distribuido tipo gossip, donde las estimaciones locales son actualizadas por estimaciones de los simplices que son adyacentes de forma superior o inferior.

Condición de conectividad entre dos simplices adyacentes

El siguiente resultado está en términos de número de saltos de comunicación (*hop-count*), los cuales son definidos como el menor número de enlaces de comunicación entre dos nodos en un SDM.

Proposición 3.3.12. Sean los k -simplices $\{\sigma_1, \sigma_2\} \in \mathcal{R}$, con $\sigma_1 \in \mathcal{R}_{j_1}$, $\sigma_2 \in \mathcal{R}_{j_2}$, y los subcomplexes $\mathcal{R}_{j_1}, \mathcal{R}_{j_2} \in \mathcal{R}$ son propiedad de los nodos j_1 y j_2 ejecutando el algoritmo 8. Además, sea $d(i, j)$ el hop-count entre dos nodos i, j del SDM. Así, pueden darse los siguientes casos:

1. Si $\sigma_1 \frown \sigma_2$, entonces $d(j_1, j_2) \leq 1$.
2. Si $\sigma_1 \smile \sigma_2$, entonces $d(j_1, j_2) \leq 2$.

Prueba. ■ Para el caso 1: Si los dos simplices son propiedad del mismo nodo, entonces $d(j_1, j_2) = 0$. Si $\sigma_1 \frown \sigma_2$, entonces ambos k -simplices pertenecen a un $k + 1$ -simplex. Dado que en un simplex dos vértices son adyacentes entre sí: $d(j_1, j_2) \leq 1$.

- Para el caso 2: Si $\sigma_1 \smile \sigma_2$, considerar cualquier par de vértices, uno por cada simplex. Este par de vértices puede estar a lo sumo un borde de distancia desde un $k - 1$ -simplex común. Así ambos nodos están a lo sumo 2 bordes de distancia entre si, por lo que $d(j_1, j_2) \leq 2$. □

Observación 3.3.13. El cálculo del k -Laplaciano combinatorio puede ser implementado a nivel nodo, usando a lo sumo dos saltos de comunicación entre nodos vecinos.

Observación 3.3.14. Se requiere la existencia de algún protocolo de ruteo que comunique paquetes de información entre dos nodos que estén a lo sumo dos saltos separados.

Observación 3.3.15. Por las observaciones 3.3.13 y 3.3.14 dadas, el cálculo Laplacianos de alto orden es una operación local esencial que necesita una pequeña sobrecarga de comunicación para la implementación sobre SDM's.

Convergencia de cobertura para el SDM

Las propiedades de convergencia de un sistema dinámico de la forma

$$\frac{\partial \omega(t)}{\partial t} = -\mathcal{L}_k \omega(t) \quad (3.12)$$

dependen de la k -ésimo grupo homológico del simplicial complex fundamental X . Este sistema ha demostrado ser generalmente semi estable, de acuerdo a alguno de los dos siguientes casos:

- Estabilidad asintótica para cualquier elección de condiciones iniciales si y solo si $H_k(X) \cong 0$. Por lo tanto, la propagación distribuida del Laplaciano es una forma de detectar si la k -homología es trivial (sin huecos).
 - Si X tiene huecos, dependiendo de la condición inicial, el sistema dinámico converge al subespacio que abarcan las clases de homología de \mathcal{L}_k . La tasa de convergencia de este sistema está controlada por el valor propio más pequeño distinto de cero de \mathcal{L}_k .
 - Si hay exactamente un hueco en la dimensión k , este método distribuido para calcular el grupo de homología permite que el sistema dinámico converja a su representación armónica.
- Global asintóticamente estable para el cálculo del primer grupo homológico:

$$\dot{x}(t) = -L_1 x(t), x(0) = x_0 \in \mathbb{R}^{n_1} \quad (3.13)$$

si y solo si $\mathcal{D} \subseteq \mathcal{A}(N)$; donde $x(t)$ es el vector de dimensión n_1 , y n_1 corresponde al número de 1 -simplices del simplicial complex.

Observación 3.3.16. Para cualquier condición inicial $x(0)$, la trayectoria $x(t)$ para todo tiempo futuro siempre converge a un punto dentro de $\ker(L_1)$. Así, la estabilidad asintótica del sistema es el indicador para obtener una homología trivial.

Observación 3.3.17. Considerar

$$x^* = \lim_{t \rightarrow \infty} x(t) \quad (3.14)$$

como un elemento en el espacio nulo de L_1 . Así, este elemento es una representación de la homología de clase de \mathcal{R}_{rb} .

Para el caso particular en que $x^* = 0 \forall x(0)$, entonces $H_1(\mathcal{R}_{rb})$ consiste en una clase trivial, por lo que \mathcal{R}_{rb} no tendrá huecos de cobertura.

Observación 3.3.18. La estructura muy específica del primer Laplaciano combinatorio garantiza que la ejecución concurrente del criterio de verificación expresado en la ecuación 4.4 a nivel nodal es, efectivamente, una regla de actualización local, por lo que el valor del estado local de un borde se actualiza utilizando estimaciones de los bordes adyacentes.

Observación 3.3.19. Por la observación 4.1.5 hay una relación entre la ejecución distribuida del criterio de verificación expresado en la ecuación 4.4 con el algoritmo de acuerdo en tiempo continuo a ser analizado en el subcapítulo 3.4, en el que se usa el gráfico Laplaciano para llegar a un consenso (un punto en $\ker(L_1)$) sobre un grafo conectado.

Criterio de estabilidad si existen más de dos huecos en \mathcal{R}_i .- Sin embargo, Muhammad y Jadbabaie muestran en [21] que si existe en la configuración topológica dos o más agujeros, este método falla para el cálculo de múltiples grupos homológicos, debido a que la magnitud de los autovectores no se relacionan con la proximidad de agujeros ni dan una relación entre los agujeros y ciclos, sin importar las condiciones iniciales elegidas. Como solución, en [21] se cita el trabajo de Kempe [1], donde se muestra un algoritmo distribuido para la descomposición espectral de la matriz de adyacencia; y ya que la representación del k -Laplaciano a nivel simplex ha sido mostrada, se aplica el algoritmo de Kempe para dar solución al cálculo de grupos homológicos que involucren más de un agujero.

La base del algoritmo de Kempe es el esquema de iteración ortogonal estándar para calcular los k vectores propios superiores de una matriz $A \in \mathbb{R}^{n \times n}$. Primero se tiene una estimación inicial para los vectores propios e_1, e_2, \dots, e_n apilados en $Q_n \in \mathbb{R}^{n \times k}$. Después de esto se calcula la factorización QR usual de AQ_n , la cual está dada por $Q_{n+1}R$, por lo que es posible invertir a R para obtener $Q_{n+1} = R^{-1}AQ_n$. Para n grande, Q_n se aproxima a los autovectores verdaderos con un pequeño error.

Suponer que la matriz A modela alguna propiedad global del SDM, en la cual las entradas A_{ij} representan relaciones a nivel local del SDM, como la matriz de adyacencia $A_{ij} \in \{0, 1\}$, es decir, la presencia o ausencia de enlaces de comunicación entre los nodos i, j . Para el Rip complez del SDM, A puede ser una matriz de adyacencia superior o inferior entre simplices.

Consideraciones:

- Cada nodo (o simplex) maneja una fila correspondiente en Q_n .
- La información de los simplices adyacentes (superiores o inferiores) puede ser obtenida con una mínima sobrecarga en de comunicación.

Por lo tanto, el cálculo $V = AQ_n$ es de forma local y realizable, pero sin la posibilidad de tener una representación local de R^{-1} , debido a que se requiere conocimiento total de R . Para estimar una solución a esta última problemática, se parte de la siguiente factorización de Cholesky:

$$K := V^T V = RQ^T QR = R^T R \quad (3.15)$$

Así, si un simplex puede estimar localmente de alguna forma la ecuación 3.15 como \hat{K} , es posible obtener una R estimada denotada por \hat{R} . Por lo tanto, se tiene la iteración local $Q_{n+1} = \hat{R}^{-1}AQ_n$.

Observación 3.3.20. Como

$$K = \sum_{\sigma} V_{\sigma}^T V_{\sigma} \quad (3.16)$$

, donde V_{σ} es una fila de V correspondiente al simplex σ , cada simplex puede solamente generar $K_{\sigma} = V_{\sigma}^T V_{\sigma}$. Así, la idea principal expuesta en [1] es aplicar el protocolo distribuido tipo gossip por promedio local para producir una estimación global de la suma $K = \sum_{\sigma} K_{\sigma}$, teniendo así una simulación determinista de un recorrido aleatorio sobre los simplices en cada paso de la factorización QR .

Observación 3.3.21. El resultado principal de Kempe es que, al usar este algoritmo gossip, la estimación converge con la probabilidad uno, esencialmente en $O(\tau_{mix} \log^2(n))$ rondas de comunicación y cálculo, donde τ_{mix} es el tiempo de mezcla del recorrido aleatorio sobre el SDM.

3.4. Protocolo de acuerdo

3.4.1. Caso estático

Generalidades

El formalismo tratado en este protocolo a nivel estados permite analizar los medios por los cuales la topología (de interconexión) de la red dicta la dinámica sobre este sistema.

Se inicia por considerar los $n \in \mathbb{Z}^+$ nodos como *unidades dinámicas* interconectadas por enlaces de intercambio de información. Así, la tasa de cambio de cada estado del nodo se asume como la suma de sus estados relativos con respecto al conjunto de sus nodos vecinos:

$$\dot{x}_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)) \quad (3.17)$$

Donde:

- $x_i \in \mathbb{R}$.- Estado escalar de nodo i .
- $i = 1, \dots, n$.
- $N(i)$.- Conjunto de nodos adyacentes o vecinos de i .

Cuando la noción adoptada de adyacencia es simétrica, el sistema puede ser representado por:

$$\dot{x}(t) = -L(\mathcal{G})x(t) \quad (3.18)$$

Donde:

- $L(\mathcal{G}) \geq 0$.- Matriz Laplaciana de los nodos que interactúan en la red no dirigida \mathcal{G} .
- $x(t) = (x_1(t), \dots, x_n(t))^T \in \mathbb{R}$.

Así, a la ecuación 3.18 se le conoce como *dinámica de acuerdo*, en la cual la tasa de cambio de cada nodo en la red involucra los estados de los nodos vecinos. Como tal estado de acuerdo es de interés, se define formalmente a continuación:

Definición 3.4.1. El conjunto de acuerdo $\mathcal{A} \subseteq \mathbb{R}^n$ es el subespacio $\text{span}\{1\}$, es decir:

$$\mathcal{A} = \{x \in \mathbb{R}^n \mid x_i = x_j, \forall i, j\} \quad (3.19)$$

Acuerdo estático en redes no dirigidas (\mathcal{G})

A continuación se presentan las condiciones de interconexión en \mathcal{G} para alcanzar el conjunto de acuerdo \mathcal{A} .

Teorema 3.4.2. Para una red distribuida estática modelada por un grafo conectado \mathcal{G} , el protocolo de acuerdo no dirigido converge al conjunto de acuerdo \mathcal{A} con una tasa de convergencia dictada por $\lambda_2(\mathcal{G})$.

Prueba. De la solución general para las dinámicas de acuerdo:

$$x(t) = e^{-L(\mathcal{G})t}x_0 = e^{-\lambda_1(\mathcal{G})t}(u_1^T x_0)u_1 + e^{-\lambda_2(\mathcal{G})t}(u_2^T x_0)u_2 + \dots + e^{-\lambda_n(\mathcal{G})t}(u_n^T x_0)u_n \quad (3.20)$$

donde:

- u_1, u_2, \dots, u_n .- Vectores propios normalizados y mutuamente ortogonales de $L(\mathcal{G})$, correspondiente a sus valores propios ordenados: $0 = \lambda_1(\mathcal{G}) < \lambda_2(\mathcal{G}) \leq \dots \leq \lambda_n(\mathcal{G})$.

se observa que, para un grafo conectado, $\lambda_i(\mathcal{G}) > 0$ (con $i \geq 2$) y $\lambda_2(\mathcal{G}) = 0$. Así:

$$x(t) \rightarrow (u_1^T x_0)u_1 = \frac{\mathbf{1}^T x_0}{n} \mathbf{1} \text{ cuando } t \rightarrow \infty \quad (3.21)$$

y por lo tanto $x(t) \rightarrow \mathcal{A}$. □

Observaciones:

- Aquí se aplica la teoría espectral de grafos relacionada con el rango de la matriz Laplaciana (o de forma equivalente, la multiplicidad por su valor propio cero) y como estas condiciones algebraicas se relacionan con la estructura del grafo.
- Como los estados de los nodos evolucionan hacia \mathcal{A} , se tiene que:

$$\frac{d}{dt}(\mathbf{1}^T x(t)) = \mathbf{1}^T(-\mathcal{L}x(t)) = -x(t)^T \mathcal{L} \mathbf{1} = 0 \quad (3.22)$$

donde $\mathbf{1}^T x(t) = \sum_i x_i(t)$ es el *centroide de los estados de la red*, evaluado para todo $t \geq 0$; es decir, es la *constante de movimiento para la dinámica de acuerdo* expresada en la ecuación 3.18. Así, la trayectoria de estado generada por el protocolo de acuerdo converge a la proyección de su estado inicial:

$$\arg \min_{x \in \mathcal{A}} \|x - x_0\| = \frac{\mathbf{1}^T x_0}{\mathbf{1}^T \mathbf{1}} \mathbf{1} = \frac{\mathbf{1}^T x_0}{n} \mathbf{1} \quad (3.23)$$

- La solución general de las dinámicas de acuerdo expresada en la ecuación 3.20 indica que, en orden de tener convergencia hacia el subespacio de acuerdo desde una condición inicial arbitraria, es necesario y suficiente tener que $\lambda_2(\mathcal{G}) > 0$.
- En el capítulo dos de [19] se muestra relación de la positividad de $\lambda_2(\mathcal{G})$ con la conectividad de \mathcal{G} , concluyendo que la estructura de orden mínima necesaria para la convergencia asintótica al acuerdo es una red interconectada que contenga un *árbol de expansión*.

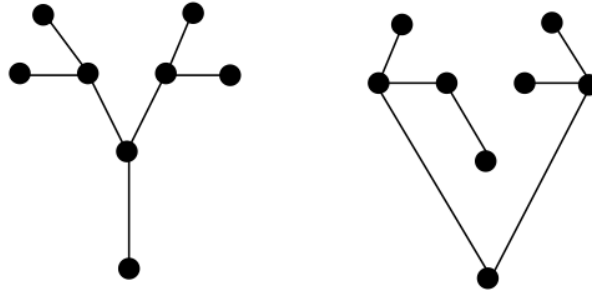


Figura 3.3: Dos ejemplos de árboles de expansión de 8 nodos. Figura tomada de [19].

Acuerdo estático en redes dirigidas (\mathcal{D})

De manera similar a las redes no dirigidas, en los digrafos (grafos dirigidos) también se puede adoptar la *dinámica de acuerdo* de la siguiente forma:

$$\dot{x}_i(t) = -L(\mathcal{D})x(t) \quad (3.24)$$

Donde:

- $L(\mathcal{D}) \geq 0$.- Matriz Laplaciana de los nodos que interactúan en la red dirigida y ponderada \mathcal{D} .
- $x(t) = (x_1(t), \dots, x_n(t))^T \in \mathbb{R}$.

El objetivo también es mostrar condiciones necesarias y suficientes sobre la interconexión de \mathcal{D} que lleven a la convergencia de sistemas de la forma 3.24 hacia el subespacio de acuerdo \mathcal{A} (ecuación 4.8). Así, se aplica el siguiente teorema para establecer un protocolo de acuerdo para una red distribuida estática modelada por un digrafo ponderado \mathcal{D} :

Teorema 3.4.3. *Para un digrafo \mathcal{D} que contenga una ramificación arraigada, la trayectoria de estado generada por la ecuación 3.24, iniciada desde x_0 , satisface:*

$$\lim_{t \rightarrow \infty} x(t) = (p_1 q_1^T) x_0 \quad (3.25)$$

donde:

- p_1, q_1 .- Los autovectores derecho e izquierdo, respectivamente, asociados al autovalor cero de $L(\mathcal{D})$, normalizado tal que $p_1^T q_1 = 1$.

En otras palabras, se tiene que $x(t) \rightarrow \mathcal{A} \forall x_0$ si y solo si \mathcal{D} contiene una ramificación arraigada.

Observaciones:

- Un digrafo \mathcal{D} tiene *ramificación arraigada* si:
 1. No contiene un ciclo dirigido.
 2. Tiene un vértice raíz v_r tal que por cada otro vértice $v \in \mathcal{D}$ hay una ruta directa desde v_r hacia v .

En la figura 3.4 a) se muestra un dígrafo que contiene una ramificación arraigada como un subgrafo. La ramificación arraigada correspondiente se da en la parte b) de la figura, junto con el nodo raíz v_r .

- Un digrafo \mathcal{D} sobre n vértices contiene un *subgrafo con ramificación arraigada* si y solo si: $\text{rank}(L(\mathcal{D})) = n - 1$. En este caso, $\mathcal{N}(L(\mathcal{D}))$ es espaneado por el vector de unos.
- Como la constante de movimiento para el protocolo de acuerdo sobre un gráfico no dirigido es la suma de los estados de nodo en un momento dado; análogamente, para identificar la cantidad conservada para el protocolo de acuerdo en evolución sobre dígrafos, se tiene la siguiente manera:

Proposición 3.4.4. *Sea q el autovector izquierdo del dígrafo en grado Laplaciano asociado con su valor propio cero. Entonces la cantidad $q^T x(t)$ permanece invariable sobre la dinámica de acuerdo 3.24.*

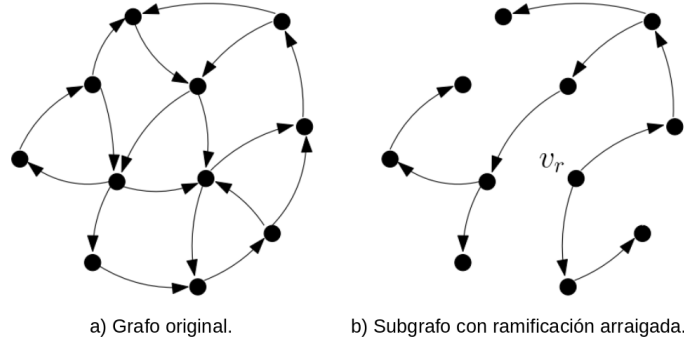


Figura 3.4: Figura tomada de [19].

Prueba. Ya que $q^T L(\mathcal{D}) = 0$, se tiene que:

$$\frac{d}{dt}\{q^T x(t)\} = -q^T \{q^T L(\mathcal{D})\} = 0$$

□

Por otro lado, el protocolo de acuerdo sobre un dígrafo que contenga una ramificación arraigada alcanza el valor promedio sobre sus estados iniciales de nodos si el autovector izquierdo de $L(\mathcal{G})$ (correspondiente a su autovalor cero) es un múltiplo escalar del vector de todos unos. En este caso, del teorema 3.4.3, $p_1 q_q^T$ se reduce a la matriz $\frac{1}{n} \mathbf{1}\mathbf{1}^T$. Así, este resultado particular introduce a la noción de *dígrafos balanceados*:

Definición 3.4.5. Un dígrafo es balanceado si, por cada vértice, los grados de entrada y salida son iguales.

Observaciones:

- Cuando el dígrafo es balanceado, además de que se cumpla que $L(\mathcal{D})\mathbf{1} = 0$, se cumple que: $\mathbf{1}^T L(\mathcal{D}) = 0$
- Así, si el dígrafo es balanceado y contiene una ramificación arraigada, entonces el valor común alcanzado por el protocolo de acuerdo es el *valor promedio de los nodos iniciales*, es decir, el *consenso promedio*, ya que:

$$\lim_{t \rightarrow \infty} x(t) = \frac{1}{n} \mathbf{1}\mathbf{1}^T x_0$$

■

Definición 3.4.6. Un dígrafo está fuertemente conectado si, entre cada par de vértices distintos, hay una trayectoria directa.

■

Definición 3.4.7. Un dígrafo está débilmente conectado si su versión no orientada es conectada.

- Se dice que un dígrafo \mathcal{D} es no orientado si todos sus bordes dirigidos están reemplazados por los bordes no dirigidos.

Así, con estas observaciones marcadas, se da el siguiente resultado para el *consenso promedio*:

Teorema 3.4.8. *El protocolo de acuerdo sobre un dígrafo alcanza el consenso promedio para cada condición inicial si y solo si está débilmente conectado y equilibrado.*

Acuerdo estático y cadenas de Markov

Se considera al proceso estocástico $X(k)$ (con $k \in \mathcal{Z}$), el cual asume uno de los n estados x_1, \dots, x_n en un tiempo dado. Además, esta cadena satisface la *propiedad de Markov*:

$$\Pr\{X(k+1) = x_j | X(k) = x_{i_k}, X(k-1) = x_{i_{k-1}}, \dots, X(0) = x_{i_0}\} = \Pr\{X(k+1) = x_j | X(k) = x_{i_k}\} \quad (3.26)$$

Así, la ecuación 3.26 permite caracterizar una *cadena de Markov* por su matriz de transición de estado P , donde su entrada p_{ij} denota la probabilidad de que la variable aleatoria X , teniendo un estado i en tiempo k , asuma un estado j en tiempo $k+1$:

$$p_{ij} = \Pr\{X(k+1) = x_j | X(k) = x_i\} \quad (3.27)$$

Además, ya que

$$\sum_{j=0}^n p_{ij} = 1$$

P es una matriz estocástica con un autovalor unitario.

Por otro lado, se define al vector de distribución de probabilidad en un tiempo k , denotado por $\pi(k)$, y cuyas i -ésimas entradas indican la probabilidad de que $X(k) = x_i$. Así, con la noción de matriz de transición se permite evaluar la evolución temporal del vector de distribución de la siguiente forma:

$$\pi(k+1)^T = \pi(k)^T P \quad (3.28)$$

De la ecuación 3.28, cuando se alcanza un valor de estado estable π^* , este es caracterizado por el autovector izquierdo de P asociado con su autovalor unitario: $\pi^* = P^T \pi^*$.

Conexión con el protocolo de acuerdo. Ya con lo anterior, es posible analizar las conexiones entre el protocolo de acuerdo sobre dígrafos ponderados (ecuación 3.24) y la evolución en tiempo discreto definida en la ecuación 3.28 por el vector de distribución de probabilidad. Para tal efecto, se evalúa el progreso de la ecuación 3.24 en δ intervalos:

$$z(k+1) = e^{-\delta L(\mathcal{D})} z(k), \text{ con } z(k) = x(\delta k) \text{ y } \delta > 0 \quad (3.29)$$

Proposición 3.4.9. Para todo dígrafo \mathcal{D} , y considerando intervalos de muestreo $\delta > 0$, se tiene que $e^{-\delta L(\mathcal{D})}$ es una matriz estocástica, es decir:

$$e^{-\delta L(\mathcal{D})} \mathbf{1} = \mathbf{1} \text{ y } e^{-\delta L(\mathcal{D})} \geq 0 \quad (3.30)$$

Así, los autovectores izquierdo y derecho de $e^{-\delta L(\mathcal{D})}$ son los de $L(\mathcal{D})$, respectivamente, asociados con los autovalores $e^{\delta \lambda_i}$, con $i = 1, \dots, n$.

Corolario 3.4.10. El estado de los nodos, durante la evolución del protocolo de acuerdo sobre \mathcal{D} , en cualquier tiempo, es una combinación convexa de los valores de todos los nodos en un tiempo previo.

La última parte para completar la correspondencia entre el protocolo de acuerdo y las cadenas de Markov refiere a no negatividad y normalización de los estados de los nodos. Para esto, se formula la proposición 3.4.11 a partir de las siguientes premisas sobre el estado:

- El estado en la cadena de Markov gobernado por la representación en 3.29 es un vector de distribución de probabilidad, es decir, pertenece a la unidad simplex.
- El estado en la dinámica de acuerdo, expresada en 3.24 es $x \in \mathbb{R}^n$

Proposición 3.4.11. *El comportamiento de la dinámica de acuerdo, representada en 3.24, es caracterizado por su acción sobre la unidad simplex.*

3.5. Restricciones para la comunicación en tiempo real

3.5.1. Comportamiento dinámico en tiempo real

Las estrategias de tiempo real están basadas en tener sistemas con respuesta acotada durante su operación dinámica. Un SISTEMA DE TIEMPO REAL (RTS) la interacción entre sus elementos es con una certeza de respuesta temporal (tanto local como global) a través de los elementos de procesamiento (procesos) [3]. Así, los sistemas distribuidos cuya funcionalidad está restringida por tiempos son conocidos como SISTEMAS DISTRIBUIDOS EN TIEMPO REAL.

Una APLICACIÓN EN TIEMPO REAL está caracterizada por un conjunto Γ de m tareas RTT asociadas $\{\tau_0, \tau_1, \dots, \tau_m\}$ que deben ser completadas para lograr un comportamiento deseado. Por esto, una tarea está compuesta por una secuencia de activaciones idénticas a_i ($i = 1, \dots, m$) separadas entre ellas por un lapso de tiempo. Así, los parámetros de una RTT son:

- Tiempo de llegada o de liberación (r_i).- Tiempo en que la tarea es establecida para su ejecución.
- Tiempo de consumo (c_i).- Tiempo que le toma a cada tarea para ejecutar sus actividades asignadas sin interrupción.
- Plazo de entrega (*deadline*) (d_i).- Límite de tiempo en que la tarea debe de finalizar. De este parámetro, se tienen las siguientes definiciones:
 - TAREA PROGRAMABLE.- Tarea que completa su objetivo en un tiempo menor o igual a d_i .
 - SISTEMA PROGRAMABLE.- RTS en que todas sus tareas terminan antes de sus deadlines.
- Tiempo de inicio (s_i).- Tiempo en que la tarea inicia su ejecución.
- Tiempo de finalización (f_i).- Tiempo en que la tarea termina su ejecución. Así, una tarea es programable si $f_i \leq d_i$.
- DESPLAZAMIENTO (φ_i).- Tiempo de la primera instancia periódica producido cuando la tarea tiene un retraso por el cual no inicie exactamente en el tiempo s_i .

Las TAREAS EN TIEMPO REAL (RTT) son entidades ejecutables de trabajo J_i en un tiempo l , que al menos son caracterizadas por un tiempo de ejecución máximo y una restricción de tiempo, y está formada por un conjunto de instancias $J_i = \{J_{i,l}\}$, con $i, l \in \mathbb{Z}^+$.

Una INSTANCIA $\{J_{i,l}\}$ es una i -ésima unidad de trabajo de una tarea, y está definida por la siguiente tupla:

$$J_{i,l} = (r_{i,l}, c_{i,l}, d_{i,l}), i, l \in \mathbb{Z}^+ \quad (3.31)$$

Cuando solo se desarrollará una sola tarea, se puede prescindir del índice i .

Capítulo 4

ALGORITMO GENERAL PARA EL SDM DESARROLLADO

A partir del análisis presentado en el capítulo 3 para la representación, existencia de protocolos que resuelvan alguna determinada tarea, la cobertura e identificación de cúmulos y las restricciones en tiempo real; se presenta en este capítulo el algoritmo distribuido generalizado a ser desarrollado de forma concurrente por cada uno de los nodos del SDM.

4.1. Algoritmo general propuesto para al SDM considerando RTT

4.1.1. Restricciones temporales

- Se considera una unidad temporal de tiempo Δd como restricción total para la ejecución de alguna tarea dada para SDM.
- A partir del tiempo $d = 0$ hasta un tiempo posterior $\frac{\Delta t}{3}$ será la primera restricción temporal para la construcción de las tuplas de simples $L_k, k = 0, 1, 2$. Por lo anterior, la instancia para esta tarea está caracterizada por la siguiente tupla:

$$J_{0,d} = (r_{0,d}, c_{0,d}, d_{0,d}) = (0, \frac{\Delta d}{3}, \leq \frac{\Delta d}{3}) \quad (4.1)$$

- El tiempo restante será destinado para la realización de alguna tarea distribuida dada, particularizando en el *consenso*.

$$J_{1,d} = (r_{1,d}, c_{1,d}, d_{1,d}) = (0, \frac{2}{3}\Delta d, \leq \Delta d) \quad (4.2)$$

4.1.2. Flujo general del algoritmo

4.1.3. Subproceso pasoMensajes

Como se muestra en la figura 4.2, en esta etapa se aplicó la condición existencia de un protocolo para la solución del paso de mensajes entre todos los nodos del SDM, es decir, sin considerar a qué cúmulo pertenezcan. Esta condición de solvencia es la especificada en el teorema 3.2.11 (detallada en

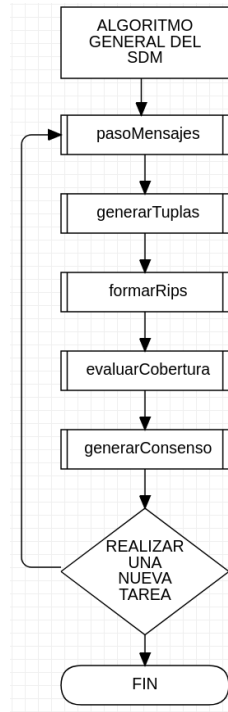


Figura 4.1: Algoritmo general propuesto para el modelo del SDM a lo largo de un tiempo Δd .

la última parte del subcapítulo 3.2.1), para la cual, de acuerdo con el bosquejo de prueba propuesto, requiere de la ejecución secuencial del protocolo $(t + 1)$ – *set agreement* y del procolo de acuerdo baricéntrico. Ya con estas ejecuciones, la condición de solvencia será satisfecha si en el nodo elegido por el $(t + 1)$ *set-agreement* se calcula la composición de estos protocolos, y esta logra generar un mapa continuo $f : |skel^t \mathcal{O}| \mapsto |\mathcal{O}|$.

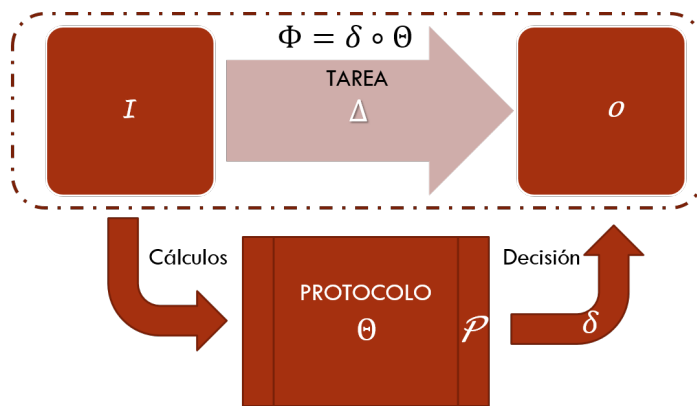


Figura 4.2: Esquema de la relación entre protocolos y tareas.

4.1.4. Subproceso generarTuplas

Al ya tener la certeza de ejecutar una tarea por el paso de mensajes, a continuación se ejecuta el subproceso generarTuplas de forma distribuida. Esta etapa consiste en las ejecuciones a nivel nodo

especificadas en el algoritmo 7 llamado encontrarSimplices() (detallado en el subcapítulo 3.3.1), por el cual serán construidas las listas L_k de simplices de orden $k = \{0, 1, 2\}$.

4.1.5. Subproceso generarRips

Después de generar las L_k listas de simplicial complexes, éstas serán necesarias para obtener una representación local de los cúmulos que conforman cada uno de los i -ésimos Rip complexes R_i . Esta etapa consiste en las ejecuciones a nivel nodo especificadas en el algoritmo 8 llamado simplexMiembro() (detallado en el subcapítulo 3.3.1).

Así, estos \mathcal{R}_i a su vez serán parte del Rip complex total \mathcal{R} dentro de una región \mathcal{D} de interés, para así describir a todo el SDM bajo el enfoque homológico simplicial.

4.1.6. Subproceso evaluarCobertura

La descomposición espectral distribuida de operadores Laplacianos de alto orden sobre Rips complexes fue el procedimiento seguido para el cómputo de grupos homológicos (con base en los resultados de Jadbabaie y Muhammad expresados en [21]) para detectar la ausencia o la existencia de huecos en la cobertura del SDM. Así, se implementó un algoritmo tipo gossip para recuperar una estimación global de ciertas matrices que no puedan ser calculadas localmente debido a huecos, y así lograr calcular una descomposición espectral distribuida de los Laplacianos.

La idea principal fue calcular el espacio nulo del primer Laplaciano combinatorio de un complejo simplicial correspondiente a la cobertura, y así extraer la información que proporciona sobre la primera homología.

Subproceso kLaplacianoCalculo

En términos matriciales de adyacencia y de grado, los $k - Laplacianos$ combinatorios se calculan de la siguiente forma:

$$L_k = D_U^{(k)} - A_U^{(k)} + (k + 1)\mathbb{I}_{nk} + A_L^{(k)} \quad (4.3)$$

, donde $A_U^{(k)}$ y $A_L^{(k)}$ son las matrices de adyacencia superior e inferior, respectivamente, y $D_U^{(k)}$ es la matriz de grado superior. Así, de esta representación combinatoria se desprenden las siguientes observaciones:

Observación 4.1.1. La ecuación 4.3 implica que la i -ésima fila de L_k depende únicamente de las interacciones locales entre el i -ésimo $k - simplex$ con sus $k - simplices$ adyacentes superiores e inferiores.

Observación 4.1.2. La ecuación 4.3 indica que los $k - Laplacianos$ trabajan de forma similar a un algoritmo distribuido tipo gossip, donde las estimaciones locales son actualizadas por estimaciones de los simplices que son adyacentes de forma superior o inferior.

Subproceso kLaplacianoEstabilidad

Desde el punto de vista de tratar el cálculo del primer Laplaciano combinatorio como un sistema dinámico:

$$\dot{x}(t) = -L_1 x(t), x(0) = x_0 \in \mathbb{R}^{n^1} \quad (4.4)$$

este será global asintóticamente estable si y solo si $\mathcal{D} \subseteq \mathcal{A}(N)$; donde $x(t)$ es el vector de dimensión n_1 , y n_1 corresponde al número de 1-*simplices* del simplicial complex.

Observación 4.1.3. Para cualquier condición inicial $x(0)$, la trayectoria $x(t)$ para todo tiempo futuro siempre converge a un punto dentro de $\ker(L_1)$. Así, la estabilidad asintótica del sistema es el indicador para obtener una homología trivial.

Observación 4.1.4. Considerar

$$x^* = \lim_{t \rightarrow \infty} x(t) \quad (4.5)$$

como un elemento en el espacio nulo de L_1 . Así, este elemento es una representación de la homología de clase de \mathcal{R}_{rb} .

Para el caso particular en que $x^* = 0 \forall x(0)$, entonces $H_1(\mathcal{R}_{rb})$ consiste en una clase trivial, por lo que \mathcal{R}_{rb} no tendrá huecos de cobertura.

Observación 4.1.5. La estructura muy específica del primer Laplaciano combinatorio garantiza que la ejecución concurrente del criterio de verificación expresado en la ecuación 4.4 a nivel nodal es, efectivamente, una regla de actualización local, por lo que el valor del estado local de un borde se actualiza utilizando estimaciones de los bordes adyacentes.

En resumen, para verificar la cobertura en un SDM, es suficiente configurar un sistema dinámico lineal distribuido expresado en la ecuación 4.4 para una condición inicial aleatoria y observar el valor del estado asintótico cuando t tiende al infinito. Casos:

1. Si 4.4 converge a cero, entonces el grado de $H_1(\mathcal{R}_{rb})$ es cero y, por lo tanto, no tiene huecos.
2. Si el valor asintótico de 4.4 no es cero para alguna condición inicial, entonces $H_1(\mathcal{R}_{rb})$ no es trivial y, por lo tanto, existe un ciclo de orden 1 no trivial en SDM y por lo tanto, la presencia de huecos en la cobertura.

4.1.7. Subproceso generarConsenso

Al ya haber determinado las condiciones de existencia de protocolos que resuelven tareas por medio del paso de mensajes mediante el enfoque topológico combinatorio, así como las condiciones de cobertura de los \mathcal{R}_i cúmulos en el SDM $\mathcal{R} \in \mathbb{R}^2$ por el enfoque homológico algebraico, finalmente se relacionan ambos enfoques tratados con el formalismo a nivel de estados dinámicos, para así analizar los medios por los cuales la topología (de interconexión) dicta la dinámica del SDM para alcanzar un subespacio $\mathcal{A} \subseteq \mathcal{R}$ correspondiente al consenso acordado.

Al retomar la observación 4.1.5, se concluye una relación entre la ejecución distribuida del criterio de verificación expresado en la ecuación 4.4 con el algoritmo de acuerdo en tiempo continuo a ser descrito a continuación, en el que se usa el gráfico Laplaciano para llegar a un consenso (un punto en $\ker(L_1)$) sobre un grafo conectado, el cual será el obtenido por el Rip complex \mathcal{R} .

Al considerar los $n \in \mathbb{Z}^+$ nodos como *unidades dinámicas* interconectadas por enlaces de intercambio de información, la tasa de cambio de cada estado del nodo se asume como la suma de sus estados relativos con respecto al conjunto de sus nodos vecinos:

$$\dot{x}_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)) \quad (4.6)$$

Donde:

- $x_i \in \mathbb{R}$.- Estado escalar de nodo i .
- $i = 1, \dots, n$.
- $N(i)$.- Conjunto de nodos adyacentes o vecinos de i .

Cuando la noción adoptada de adyacencia es simétrica, el sistema es conocido como *dinámica de acuerdo* y es representado por:

$$\dot{x}_i(t) = -L(\mathcal{G})x(t) \quad (4.7)$$

Donde:

- $L(\mathcal{G}) \geq 0$.- Matriz Laplaciana de los nodos que interactúan en la red no dirigida \mathcal{G} .
- $x(t) = (x_1(t), \dots, x_n(t))^T \in \mathbb{R}$.

Definición 4.1.6. El conjunto de acuerdo $\mathcal{A} \subseteq \mathbb{R}^n$ es el subespacio $\text{span}\{1\}$, es decir:

$$\mathcal{A} = \{x \in \mathbb{R}^n \mid x_i = x_j, \forall i, j\} \quad (4.8)$$

Subproceso consensoG

La siguiente condición de convergencia del SDM hacia un consenso dictado por el subconjunto \mathcal{A} aplica si la representación del SDM por medio del Rip complex \mathcal{R} es no dirigido, es decir, asumiendo comunicación bidireccional de los nodos.

Teorema 4.1.7. *Para un SDM en un instante de tiempo dado, modelado por un Rip complex conectado no dirigido \mathcal{R} , el protocolo de acuerdo no dirigido converge al conjunto de acuerdo \mathcal{A} con una tasa de convergencia dictada por $\lambda_2(\mathcal{R})$.*

Así, la solución general de las dinámicas de consenso expresada en la ecuación 4.9

$$x(t) = e^{-L(\mathcal{R})t}x_0 = e^{-\lambda_1(\mathcal{R})t}(u_1^T x_0)u_1 + e^{-\lambda_2(\mathcal{R})t}(u_2^T x_0)u_2 + \dots + e^{-\lambda_n(\mathcal{R})t}(u_n^T x_0)u_n \quad (4.9)$$

indica que, en orden de tener convergencia hacia el subespacio de acuerdo desde una condición inicial arbitraria, es necesario y suficiente tener que $\lambda_2(\mathcal{G}) > 0$, lo cual implica que la estructura de orden mínima necesaria para la convergencia asintótica al acuerdo es una red interconectada que contenga un *árbol de expansión*.

4.2. Algoritmo a nivel nodal orientado a eventos

En este subcapítulo se detallan los estados, eventos y datos a ser almacenados por cada uno de los nodos dentro del SDM.

- Durante la ejecución del algoritmo, cada nodo puede estar en uno de los cuatro estados: *iniciar*, *esperar*, *crítico* y *actualizar*:
 - **Iniciar.**- Después de completar este estado, un nodo debe describirse como un nodo de espera, un nodo crítico o un nodo de actualización.

- **Esperar.**- Un nodo en espera establece su temporizador de muestra para los siguientes casos:
 - Cuando se dispara el temporizador de muestra, el nodo sondea el entorno a través de un sensor designado por el protocolo de acuerdo, y entra en estado CRÍTICO si el estado del sensor ha aumentado por encima de alguna restricción temporal.
 - Si el estado del sensor está por debajo del umbral, el nodo restablece su temporizador de muestra y permanece en estado de ESPERA.
 - Si un nodo en espera recibe un mensaje de solicitud de datos, y si el estado de su sensor relativo al umbral es alto, transmite un mensaje de datos al nodo crítico y establece un temporizador de respuesta.
 - Si se dispara el temporizador de respuesta, este mensaje se reenvía.
 - Si un nodo en espera recibe un mensaje de actualización, transmite, recibe el ACK y almacena los datos si el estado de su nodo en relación con el umbral es alto.
 - **Crítico.**- El nodo crítico transmite una solicitud de datos a cada vecino. A medida que los vecinos responden, el nodo crítico construye sus arreglos de simplices de orden 0 y 1. Para cada componente existente único que contribuye al nuevo componente, 0-simplices, 1-simplices y el Rip complex se actualizan adecuadamente. Después de calcular las condiciones de solvencia mediante el paso de mensajes, de cobertura y de convergencia hacia un subconjunto \mathcal{A} para la realización de consenso, el nodo crítico pasa el estado ACTUALIZAR.
 - **Actualizar.**- Un nodo de actualización inicia un temporizador de respuesta y difunde un mensaje de actualización a cada uno de sus vecinos. Si no se recibe un ACK para un nodo receptor particular, el mensaje se reenvía. Si se recibe un mensaje de actualización, el nodo ACKs el nodo de transmisión. Después de que el nodo de actualización haya recibido ACK de todos sus vecinos, el nodo pasa al estado de ESPERA.
- Solo un nodo puede estar en su estado CRÍTICO en cualquier etapa de la ejecución de un algoritmo.
 - Pueden ocurrir tres tipos de eventos (interrupciones) en cada nodo:
 - Disparos de temporizador de muestra.- Indica que el entorno debe muestrearse a través de un sensor designado , por la previa ejecución del protocolo de consenso.
 - Disparos de temporizador de respuesta.- Indica que un mensaje debe reenviarse.
 - Mensaje de recepción.- Indica que se ha recibido un mensaje.
 - Para cada evento hay tareas complementarias, las cuales permiten que un nodo responda a un evento o establezca un disparador para habilitar o deshabilitar un evento potencial: iniciar/detener el temporizador de muestra, iniciar/detener el temporizador de respuesta y transmitir/retransmitir el mensaje.
 - Los datos de nodo y la actualización contienen los siguientes elementos de datos: listas L_k , las m -instancias en instantes de tiempo d $J_{m,d}$; así como el conjunto de valores Val con los resultados de las condiciones de resolución de tareas por paso de mensajes, resultados de las condiciones de cobertura, así como los resultados del subconjunto \mathcal{A} a converger para la solución del consenso.

Capítulo 5

CONCLUSIONES Y TRABAJO A FUTURO

5.1. Conclusiones

- Primeramente se logró aplicar el enfoque topológico combinatorio para determinar la existencia o no de algún protocolo que resuelva alguna tarea incolora mediante el paso de mensajes, con base a los trabajos de Herlihy, Rajsbaum y Kozlov. Esta forma de abordar el problema permitió abstraer la dinámica de los procesos en objetos algebraicos combinatorios llamados simplicial complex abstractos, los cuales capturan las propiedades esenciales de la dinámica del SDM por su relación con invariantes topológicos.
- Por la aplicación de herramientas de la topología simplicial algebraica, se logró construir analíticamente una representación de los procesos y sus enlaces de comunicación mediante Rips complexes, los cuales son una muy buena aproximación hacia los simplicial complex abstractos; y por el cálculo de sus grupos homológicos $H_k(\mathcal{R}_i)$ vía operadores laplacianos, se permitió evaluar las condiciones de cobertura de los cúmulos formados en el SDM cuando sus nodos tienen poca o nula capacidad de procesamiento geográfico (GPS, por ejemplo). Para lograr tal representación, se tuvo como base los trabajos de Ghrist, De Silva y Muhhamad.
- Se relacionó el enfoque homológico simplicial algebraico con la representación dinámica de un sistema lineal representado por un Rip complex \mathcal{R} no dirigido, para así retomar las condiciones de estabilidad y convergencia del SDM hacia un subconjunto de consenso estudiadas por Mesbahi, Murray y Olfati.
- Se logró la aplicación de estos tres enfoques al proponer un algoritmo genérico a ser implementado en un SDM con acotaciones en tiempo real, destacando los eventos y estados de cada uno de los agentes involucrados en el SDM.

5.2. Trabajo a futuro

- Se han dado las bases algorítmicas a nivel nodo y de forma distribuida para ser implementados en la plataforma de pruebas con la que cuenta el laboratorio de SISTEMAS DE CÓMPUTO MÓVIL dentro del IIMAS-UNAM. Los nodos de dicha plataforma se conforman por ordenadores Raspberry Pi 3 modelo B, así como plataformas de prueba tipo arduino Yun.

- Refinar el procedimiento para el cálculo de la cobertura total del SDM, para así prescindir de nodos barrera que delimiten el área total a ser modelada por los \mathcal{R}_i cúmulos, ya que esta es una restricción que pocas veces se cumple en SDM de gran escala.
- Refinar el procedimiento de cálculo de grupos homológicos, para así lograr estimar la existencia y ubicación de más de dos huecos de comunicación dentro de los cúmulos.
- Particularizar los resultados para la existencia de protocolos que resuelvan diferentes tipos de tareas (coloras en incoloras).

Apéndice A

Base de la topología combinatorial: complexes

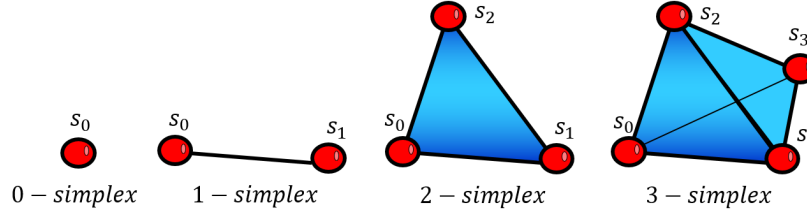
Para la representación de los simplicial complex se tienen dos maneras: topológica combinatorial y realización geométrica; las cuales son análogas en algunos de sus elementos. Para la descripción estos tales elementos, se respetó parte de la terminología del inglés, ya que en este idioma se reconocen mejor en la literatura.

Definición A.0.1. Dado un conjunto finito \mathcal{S} y una familia \mathcal{A} de subconjuntos finitos de \mathcal{S} , se dice que \mathcal{A} es un **simplicial complex abstracto** en \mathcal{S} si se satisfacen las siguientes condiciones:

- Si $\mathcal{X} \in \mathcal{A} \wedge \mathcal{Y} \subseteq \mathcal{X}$, entonces $\mathcal{Y} \in \mathcal{A}$.
- $v \in \{\mathcal{A}\} \forall v \in \mathcal{S}$.

De la definición A.0.1 se desprenden los siguientes elementos:

- *Vértices.* Elementos denotados por letras minúsculas: s, v, x, \dots
 - Forma combinatoria.- un elemento de un conjunto: $s, v, x, \dots \in \mathcal{S}$.
 - Forma geométrica.- un punto en un espacio Euclidiano \mathbb{R}^d .
- *Simplices.*- Elementos denotados por letras minúsculas griegas: $\sigma, \tau, \dots \in \mathcal{A}$.
 - Forma combinatoria.- un conjunto de vértices. Se dice que un simplex $\sigma \in \mathcal{A}$ tiene *dimensión* $|\sigma| - 1$, donde $|\sigma|$ denota la cardinalidad de σ . Como se muestra en la figura A.1, un simplex de dimensión n es llamado n -*simplex*, y sus elementos se denotan por $\sigma = \{s_0, \dots, s_n\}$; en particular los vértices son simplexes de dimensión 0.
 - Forma geométrica.- una *envolvente (o casco) convexa* de puntos en posición general.
- *Simplicial complex.* Se trata de una generalización de los grafos (ya que los grafos son de dimensión 0 o 1, y los simplicial complexes son de dimensión arbitraria).
 - Forma combinatoria.- es un conjunto de simplexes.
 - Forma geométrica.- son simplexes “pegados juntos” a lo largo de sus caras.
- Sean los simplexes τ, σ , se dice que τ es un *face (cara)* de σ si $\tau \subseteq \sigma$, y que es un *face propio* si $\tau \subset \sigma$.

Figura A.1: Realizaciones geométricas de k -simplexes.

- Si τ tiene dimensión $k \implies \tau$ es un k -face de σ .
- Sea $\sigma = \{s_0, \dots, s_n\}$, se define como $Face_i\sigma$ al i -ésimo face de σ a ser el $(n-1)$ -simplex $\{s_0, \dots, \hat{s}_i, s_n\}$, donde el elemento marcado con el circunflejo (\hat{s}_i) denota omisión.
- Un complex \mathcal{B} es un *subcomplex* de \mathcal{A} si cada simplex de \mathcal{B} es también un simplex de \mathcal{A} .
- Para un n -dimensional simplex σ , se denota por 2^σ el complex que contiene σ y todos sus faces; además se denota por $\partial 2^\sigma$ al complex de faces de σ de dimensión a lo sumo $n-1$.
- Un *mapa de vértices* $\mu: V(\mathcal{A}) \rightarrow V(\mathcal{B})$ lleva cada vértice de \mathcal{A} hacia un vértice de \mathcal{B} .

Definición A.0.2. Dado un simplicial complex abstracto \mathcal{A} , se define un **simplicial complex geométrico** (denotado por $|\mathcal{A}|$) a la colección de simplices geométricos en \mathbb{R}^d tal que se complan las siguientes condiciones:

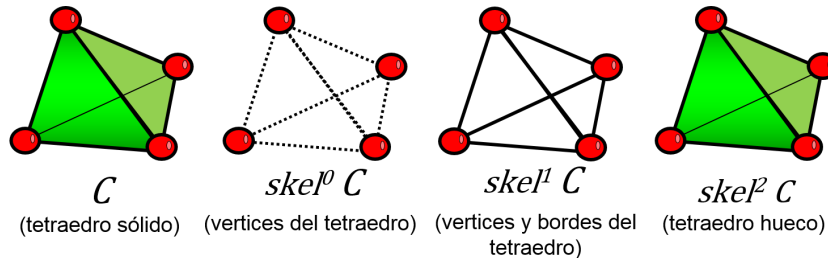
- Cualquier *face* de un $\sigma \in \mathcal{A}$ está también en \mathcal{A} .
- $\forall \sigma, \tau \in \mathcal{A}$, su intersección $\sigma \cap \tau$ es un *face* de cada uno de ellos.

Observación A.0.3. En un significado estricto, un simplicial complex abstracto es una generalización de un grafo, ya que se captura relaciones de orden superior entre los vértices de un conjunto V .

A.1. Construcciones comunes de los simplicial complex

A partir de un $\sigma \in \mathcal{C}$ se describen las siguientes construcciones estándar para un simplex.

- Sea \mathcal{C} un simplicial complex abstracto, y $l \in \mathbf{Z}^+$. El conjunto de simplices de \mathcal{C} de dimensión a lo sumo l es un subcomplex de \mathcal{C} , denominado l -skeleton y denotado por $skel^l(\mathcal{C})$. En la figura A.2 se muestran ejemplos de l -skeleton, con $l = \{0, 1, 2\}$.

Figura A.2: Realizaciones geométricas de l -skeletons.

- Un simplex σ en un complex \mathcal{C} es un *facet* si σ no es un face propio de cualquier otro simplex en \mathcal{C} . Además, como la dimensión de un complex \mathcal{C} es la máxima dimensión de algún de sus facets, por lo tanto, un *facet* de \mathcal{C} es el simplex σ de máxima dimensión.
 - Por otra parte, un complex es *puro* si todos sus facets tienen la misma dimensión.
- **Construcción en estrella** $St(\sigma, \mathcal{C})$.- Corresponde a un subconjunto de \mathcal{C} cuyos *facets* son los simplices de \mathcal{C} que contienen σ . Su realización geométrica es llamada *estrella de σ* .
 - Una *estrella abierta* $St^\circ(\sigma)$ corresponde a la unión de los simplices interiores contenidos en σ : $St^\circ(\sigma) = \bigcup_{\tau \supseteq \sigma} Int\tau$. Este objeto es un espacio topológico abierto en \mathcal{C}
- **Construcción en enlace** $Lk(\sigma, \mathcal{C})$.- Corresponde al subcomplex de \mathcal{C} construido por todos los simplices en $St(\sigma, \mathcal{C})$ que no tienen vértices comunes con σ . Su realización geométrica es llamada *enlace de σ* . En la figura A.3 se muestran, a partir de un complex \mathcal{C} de forma pentagonal, ejemplos de *Facet*, *St*, *St^o* y *Lk*.

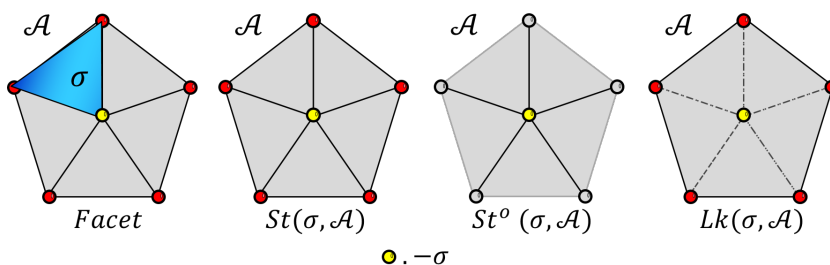


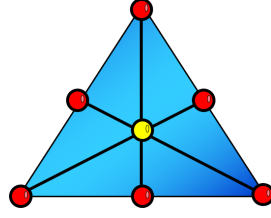
Figura A.3: Realizaciones geométricas de construcciones *Facet*, *St*, *St^o* y *Lk*.

- **Construcción en unión**.- A partir de \mathcal{A} , \mathcal{B} , $V(\mathcal{A})$ y $V(\mathcal{B})$. Su *unión* $\mathcal{A} * \mathcal{B}$ corresponde al complex con el conjunto de vértices $V(\mathcal{A}) \cup V(\mathcal{B})$ cuyos simplices son todas las uniones $\alpha \cup \beta$, con $\alpha \in \mathcal{A}$ y $\beta \in \mathcal{B}$, y ya sea α o β pueden ser conjuntos vacíos. Así, $\{\mathcal{A}, \mathcal{B}\} \subset \mathcal{A} * \mathcal{B}$; y esta construcción es conmutativa y asociativa.
- **Subdivisión baricéntrica** *Bary \mathcal{K}* .- Se trata de tipo de construcción ampliamente usada en la topología combinatoria. Como se observa en la figura A.4, la construcción geométrica es a partir de un complex \mathcal{K} . Así, el complex *Bary \mathcal{K}* es construido de forma inductiva sobre los *skeletons* de \mathcal{K} .

Se inicia tomando los vértices de \mathcal{K} . Posteriormente se inserta un baricentro en cada borde de \mathcal{K} y se toman conos (con vértices en los baricentros). En general, para extender la subdivisión baricéntrica desde el $(n - 1)$ - *skeleton* al n - *skeleton* de \mathcal{K} , se inserta un baricentro b en cada simplex $\sigma \in \mathcal{K}$, y este toma un cono con vértice sobre *Bary $\partial\sigma$* , la frontera ya subdividida de σ .

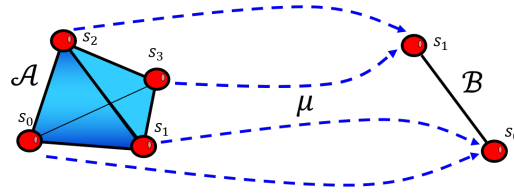
A partir de la construcción geométrica, se tiene la siguiente definición combinatoria:

Definición A.1.1. Sea \mathcal{A} un simplicial complex abstracto. Su subdivisión baricéntrica *Bary \mathcal{A}* es el simplicial complex cuyos vértices son los simplices no vacíos de \mathcal{A} . Una $(k+1)$ -*tupla* $(\sigma_0, \dots, \sigma_k)$ es un simplex de *Bary \mathcal{A}* si y solo si la tupla puede ser indexada de modo que $\sigma_0 \subset \dots \subset \sigma_k$.

Figura A.4: Realización geométrica de un $Bary(\sigma)$.

A.2. Mapeos

Definición A.2.1. Sean \mathcal{A} y \mathcal{B} dos simplicial complexes. Como se muestra en la figura A.5, el mapa de vértices μ es definido como **mapa simplicial** si este lleva simplexes a simplexes: $\mu : \mathcal{A} \mapsto \mathcal{B}$. Es decir, si $\{s_0, \dots, s_n\}$ es un simplex de \mathcal{A} , entonces $\{\mu(s_0), \dots, \mu(s_n)\}$ es un simplex de \mathcal{B} .

Figura A.5: Mapa simplicial $\mu : \mathcal{A} \mapsto \mathcal{B}$.

Definición A.2.2. Dados $\{\mathcal{A}, \mathcal{B}\}$, un mapa portador Φ de \mathcal{A} a \mathcal{B} lleva $\sigma \in \mathcal{A}$ a un subcomplex $\Phi(\sigma) \in \mathcal{B}$ tal que para todo $\{\sigma, \tau\} \in \mathcal{A}$ (tal que $\sigma \subseteq \tau$), se tiene que $\Phi(\sigma) \subseteq \Phi(\tau)$.

Se emplea notación de potencia para describir el rango y dominio de este mapa: $\Phi : \mathcal{A} \mapsto 2^{\mathcal{B}}$. Además, este mapa es monotónico, ya que: $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma) \cap \Phi(\tau)$.

Definición A.2.3. Sea el mapa portador $\Phi : \mathcal{A} \mapsto 2^{\mathcal{B}}$.

- Φ es llamado *mapa rígido* si por cada $\sigma \in \mathcal{A}$ de dimensión d , el subcomplex $\Phi(\sigma)$ es puro y de dimensión d . Además, $\sigma \in \mathcal{A}$ es no vacío si y solo si σ es no vacío.
- Φ es llamado *mapa estricto* si $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma) \cap \Phi(\tau)$, $\forall \sigma, \tau \in \mathcal{A}$. Así, hay un único $\sigma \in \mathcal{A}$ de menor dimensión, tal que $\tau \in \Phi(\sigma)$, donde σ es llamado *portador* de τ .

Definición A.2.4. Sean dos mapas portadores $\Phi : \mathcal{A} \mapsto 2^{\mathcal{B}}$ y $\Psi : \mathcal{A} \mapsto 2^{\mathcal{B}}$, así como el mapa simplicial $\varphi : \mathcal{A} \mapsto \mathcal{B}$. Se dice que:

- Φ es portado por Ψ , y denotado por $\Phi \subseteq \Psi$ si $\Psi(\sigma) \supseteq \Phi(\sigma)$, para todo $\sigma \in \mathcal{A}$.
- φ es portado por Φ si $\varphi(\sigma) \in \Phi(\sigma)$ para todo $\sigma \in \mathcal{A}$.

A.3. Configuraciones aplicadas de los simplicial complexes

[12, cap. 2] Las configuraciones a continuación presentadas son empleadas como aproximaciones a variedades, para proveer modelos de datos, así como espacios que comparan estados. Las aplicaciones de estas configuraciones van desde la asignación de trayectorias para nodos móviles en un sistema distribuido (para realizar tareas de decisión), hasta el modelado de cadenas de proteínas.

A.3.1. Complexes de estrategia e incertidumbres

[12, pgs. 34-35] Además de proveer modelos de datos, los complexes tienen aplicación como espacios que comparan estados. Para esto, se parte de un *grafo de transición*, que es un grafo dirigido X cuyos vértices V representan estados de un sistema abstracto, y cuyos bordes E representan acciones deterministas.

Una enfoque para el modelado de *incertidumbre adversa en grafos de transición* es por medio de bordes ramificados. Aquí, las acciones son elegidas de forma determinista, cuyo resultado es uno entre varias posibilidades, fuera del control del seleccionador. Estas posibilidades no son necesariamente elegidas probabilísticamente, ya que un *adversario* puede determinar cualquier resultado entre los bordes terminales. El estado elegido por el adversario es llamado *game-player*, pero solamente después que la acción no determinista ha sido ejecutada. En este escenario, se presenta la idea de una estrategia garantizada para alcanzar un *estado objetivo* a pesar de la intromisión de un adversario.

La planificación para X es directa: encontrar una trayectoria directa desde vértices iniciales hacia vértices meta, y entonces ejecutar acciones apropiadas de forma secuencial. Por otro lado, para dar solución a la planificación de trayectorias para un grafo de transición no determinista X , se aplican estructuras de datos en forma simplicial.

Definición A.3.1. Un **complex estratégico** Δ_X es un simplicial complex abstracto definido sobre el conjunto de acciones $E(X)$, en el cual un k -simplex es una colección de $k + 1$ acciones disjuntas cuyas uniones (incluyendo todas las ramas de cualquier acción no determinista elegida) no contengan ciclos dirigidos. Se cumple que Δ_X es:

- Cerrado bajo restricción.- Una colección más pequeña de tales bordes también contiene ciclos no dirigidos.
- Sensible.- Ya que un ciclo dirigido significa que un adversario hábil puede “ciclar” al jugador a través de estados indefinidamente.

Así, cada simplex de Δ_X es una estrategia para “progresar” a uno o más estados. Para un estado objetivo $g \in V(X)$, se dice que X tiene una estrategia completa para un g logrado si existe un mapeo de $V(X)$ a $E(X)$ que asocie a v cada acción basada en v , tal que se llega a g con la ejecución de al menos N acciones suficientes.

A.3.2. Espacios de configuración de gráficos discretizados

[12, pgs. 37-38] La estrategia para el problema de asignación en un sistema distribuido puede ser interpretado como un *orden simplicial de espacio de configuración*. Considerar un grafo finito X , interpretado como la geometría de una trayectoria (como el seguimiento de robots o *tokens*). El espacio de configuración de n puntos etiquetados sobre X , denotado por $\mathcal{C}^n(X)$, es relevante para estrategias de seguimiento de tokens; sin embargo, no se trata en general de una variedad ni de un simplicial/cell complex. Por lo tanto, se tiene la siguiente aproximación:

Definición A.3.2. Un *espacio de configuración discretizado* de X se define como:

$$\mathcal{D}^n(X) := (X \times \dots \times X) - \tilde{\Delta} \tag{A.1}$$

donde $\tilde{\Delta}$ denota el conjunto de todas las celdas abierta en X^n cuyos cierres cruzan la diagonal topológica Δ .

Igualmente, $\mathcal{D}^n(X)$ es el conjunto de configuraciones para las cuales, dados dos tokens sobre X y alguna trayectoria en X que los conecte, la trayectoria contiene al menos un borde entero abierto. Así, en lugar de restringir tokens a estar en al menos alguna distancia intrínseca de separación ϵ , se restringe a estar al menos a un borde completo de separación. Se remarca que $\mathcal{D}^n(X) \subset \mathcal{C}^n(X)$, ya que no contiene celdas parciales que surjan cuando se corta a lo largo de la diagonal; además $\mathcal{D}^n(X)$ es el subcomplex más largo de X^n que no cruza Δ .

Ya con estas interpretaciones, se puede hacer la siguiente caracterización:

- 0 – *cells* de $\mathcal{D}^n(X)$.- Configuraciones discretizadas como arreglos de tokens etiquetados en los vértices del grafo.
- 1 – *cells* de $\mathcal{D}^n(X)$.- Indica cual de las configuraciones discretas puede ser obtenida por el movimiento de un token a lo largo de un borde de X .
- 2 – *cells* de $\mathcal{D}^n(X)$.- Representan físicamente dos bordes independientes: se permite el movimiento de un par de tokens de forma independiente a lo largo de los bordes disjuntos.
- k – *cells* de $\mathcal{D}^n(X)$.- Representa la habilidad de mover k tokens a lo largo de k bordes de cierre disjuntos en X de una forma independiente.

A.3.3. Complexes de estados

Definición A.3.3. Dado un grafo X y una \mathcal{A} para etiquetar los vértices V por medio de un estado $u (u : V \mapsto \mathcal{A})$, un **sistema reconfigurable** es una colección de estados u_α , el cual es cerrado sobre las acciones de un conjunto fijo de reconfiguraciones locales llamadas **generadores**. Cada generador ϕ consiste de un *subgrafo de soporte* $supp_\phi \subset X$, así como de un par no ordenado de estados locales, etiquetado del conjunto de vértices de $supp_\phi$ por elementos de \mathcal{A}

Se dice que un generador es *admissible* en un estado dado u si uno de los estados locales del generador empareja la restricción de u a $supp_\phi$. Además, la *traza* de un generador es el conjunto no vacío de soporte: $trace_\epsilon \subset supp_\phi$, sobre el cual los estados locales difieren.

Apéndice B

Homología

Este es un breve resumen de las herramientas homológicas, asumiendo los siguientes fundamentos de álgebra (superior y lineal) y topología.

De álgebra:

- Anillos (generalizaciones algebraicas de \mathbb{Z}).
- Módulos (generalizaciones algebraicas de espacios vectoriales).
- Homomorfismos (generalizaciones algebraicas de transformaciones lineales).

De topología:

- Formas combinatoriales y realizaciones geométricas de simplicial complexes.

B.0.1. Generalización de los conceptos de adyacencia y grado a partir de la teoría de grafos estándar

Una propiedad clave de la definición de un simplicial complex X es que el conjunto potencia sobre el cual X es definido es cerrado bajo la formación de subconjuntos; es decir, cada *subsimplex* (face) es también un simplex en X .

Orientación en simplices

Sea un conjunto finito de vértices $B = \{b_1, b_2, \dots, b_k\}$. Un *orden arbitrario de vértices* $\{b_{\sigma_0}, b_{\sigma_1}, \dots, b_{\sigma_k}\}$ de un simplex define un k -simplex orientado, denotado por $[b_{\sigma_0}, b_{\sigma_1}, \dots, b_{\sigma_k}]$, por lo que un simplicial complex X es orientado si todos los simplices en X son orientados. En contraste, un *simplex no orientado* se denota por $\langle b_{\sigma_0}, b_{\sigma_1}, \dots, b_{\sigma_k} \rangle$. Ejemplos de simplices orientados son mostrados en la figura B.1:

Adyacencia entre k -simplices

Dados los k -simplices $\sigma, \sigma_i, \sigma_j \in X$, se tienen las siguientes definiciones:

- σ_i y σ_j son *adyacentes superiores* si ambos son *faces* de algún $(k+1)$ -simplex $\in X$. Se denota esta adyacencia por $\sigma_i \sim_U \sigma_j$ o por $\sigma_i \frown \sigma_j$. Esta última denotación (\frown) será la adoptada para el análisis y desarrollo del modelo en esta tesis.

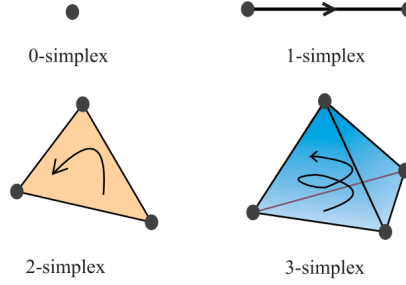


Figura B.1: Ejemplos de orientaciones de 0-, 1-, 2- y 3- *simplex*.

- Así, la representación matricial de las relaciones entre diferentes k -*simplices* es denotada por $A_U^{(k)}$.

Observación B.0.1. $A_U^{(0)}$ coincide con la típica noción de matriz de adyacencia de un grafo, la cual captura su 1-*skeleton*.

- El *grado superior* de un σ , denotado por $deg_U(\sigma)$, es el número de $(k+1)$ -*simplices* $\in X$ de los cuales σ es un *face*.
- Si σ_i y σ_j son orientados, adyacentes superiores y tienen un $(k+1)$ -*simplex* τ común, se dice que σ_i y σ_j son *orientados de forma similar* si sus orientaciones coinciden con las inducidas de σ .
- σ_i y σ_j son *adyacentes inferiores* si ambos tienen un *face* en común. Se denota esta adyacencia por $\sigma_i \sim_L \sigma_j$ o por $\sigma_i \smile \sigma_j$. Esta última denotación (\smile) será la adoptada para el análisis y desarrollo del modelo en esta tesis.
 - Así, la representación matricial de las relaciones entre diferentes k -*simplices* es denotada por $A_L^{(k)}$.
- El *grado inferior* de un σ , denotado por $deg_L(\sigma)$, es el número de *faces* $\in \sigma$.

B.0.2. Cadenas complexes como objetos fundamentales de la homología

Por la implicación del álgebra lineal como forma de construcción, la homología es un esquema de compresión de las características topológicas esenciales de una clase particular de estructuras de datos (generadas de los espacios topológicos), por lo que la homología algebraica es usualmente construida con módulos sobre un anillo conmutativo. Como forma clara de presentación, inicialmente se restringe al entorno casi trivial de espacios vectoriales finito-dimensionales sobre un campo \mathbb{F} , ya sea \mathbb{R} o \mathbb{F}_2 (± 1).

Cadenas complexes .- Sea $C_k(X)$ (para $q \geq 0$) el espacio vectorial cuyas bases son el conjunto de todos los k -*simplices* de un simplicial complex orientado X , entonces sus elementos son las *cadenas* o combinaciones lineales de los vectores base, y están denotados por k -*chains*.

Observación B.0.2. Las k -*chains* pueden ser expresadas como una suma forma finita expresada en la ecuación B.1:

$$\sum_j \alpha_j \sigma_j^{(k)} \quad (\text{B.1})$$

donde $\alpha_j \in \mathbb{R}$ y $\sigma_j^{(k)} \in X$ son los k -*simplices* orientados.

Observación B.0.3. En estas bases, un cambio en el signo del coeficiente corresponde a un cambio la orientación, como se muestra en la ecuación B.2:

$$[s_o, \dots, s_i, \dots, s_j, \dots, s_k] = -[s_o, \dots, s_i, \dots, s_j, \dots, s_k]; \text{ con: } k > j > i \quad (\text{B.2})$$

Observación B.0.4. Para una k más grande que la dimensión de X , se establece que $C_k(X) = 0$.

Así, con estas ideas fundamentales, se presentan las siguientes definiciones formales:

Definición B.0.5. Un mapa **frontera** es una transformación lineal $\partial_k : C_k(X) \mapsto C_{k-1}$, con $0 \leq k \leq \dim(X)$, que actúa sobre los elementos base $[s_0, \dots, s_k]$ de la siguiente forma:

$$\partial < s_0, \dots, s_k > := \sum_{i=0}^k (-1)^i < s_0, \dots, s_{i-1}, s_{i+1}, \dots, s_k > \quad (\text{B.3})$$

Observación B.0.6. En otras palabras, por la definición B.0.5 se mapean los elementos base ($k - chains$) hacia sus *faces* frontera, como una suma abstracta de bases $(k - 1) - simplices$, cada una con coeficientes \pm (dependiendo de su paridad). Por ejemplo:

- La frontera de una trayectoria directa en un grafo (una cadena orientada $1 - chain$) corresponde a la diferencia entre sus dos puntos finales.
- La acción del operador de límite en un 3-simplex y sus subsimplices es mostrado en la figura B.2:

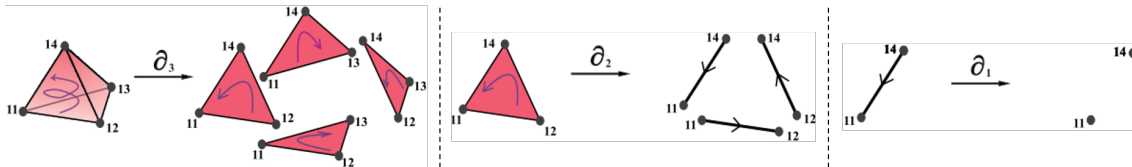


Figura B.2: Aplicación de un mapa frontera sobre 3-, 2- y 1 - simplices.

Observación B.0.7. Ya que $C_k(X)$ es espacio vectorial de dimensión finita para toda k , ∂_k tiene como representación matricial $B_k \in \mathbb{R}^{n_{k-1} \times n_k}$, con n_k como el número de $k - simplices$ de X .

En particular, la representación matricial para el primer mapa frontera ∂_1 no es más que la matriz de incidencia bordes-vértices de un grafo, es decir, el mapeo de bordes ($1 - simplicies$) a vértices ($0 - simplicies$).

Observación B.0.8. Del la definición B.0.5 surge el siguiente lema.

Lema B.0.9. *La frontera de una frontera es nula:*

$$\partial^2 = \partial_{k-1} \circ \partial_k = 0, \forall k \text{ o } \partial^2 = \partial_k \circ \partial_{k+1} = 0, \forall k \geq 1 \quad (\text{B.4})$$

Definición B.0.10. Dado un simplicial complex X , al fijar un campo \mathbb{F} , se le llama **cadena complex** $C = (C_*, \partial_*)$ a la secuencia de espacios vectoriales y transformaciones lineales, presentada de la siguiente forma:

$$\cdots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \cdots \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0 \quad (\text{B.5})$$

cumpliendo además con el lema B.0.9.

B.0.3. Características de la homología

La homología tiene tres ingredientes principales para su caracterización:

- Cadenas.- Son los objetos a ser contados.
- Grado.- Referido al tamaño o dimensión de las cadenas.
- Límite.- Referido a la cancelación de pares de cadenas de grado incidente, denotado como $\partial^2 = 0$ (“la frontera de una frontera es el espacio nulo”).

En la tabla mostrada en la figura B.3, además de la homología simplicial, se resumen las características de esta y otros dos tipos de homologías: celular y singular. Además, se destaca el resultado mostrado en el teorema B.0.11:

Teorema B.0.11. $\mathcal{H}_*^{sing} \cong \mathcal{H}_*^{cell}$ siempre que el espacio tenga una estructura bien definida (celular o simplicial).

Tipo de homología	<i>Simplicial</i> H_*^{simp}	<i>Celular</i> H_*^{cel}	<i>Singular</i> H_*^{sing}
Características			
Conjunto base	X - Simplicial complex	X - Complex celular construido por varias dimensiones, en alguna forma que admita una noción razonable de “pegado” inductivo de los complex	X - Cualquier espacio topológico
Cadenas	$C_*^{simp}(X)$ generado por simplices orientados de X	$C_*^{cel}(X)$ generado por celdas orientadas de X	$C_*^{sing}(X)$ generado por mapas σ
Grado	Dimensión de X	Dimensión de X	Dimensión del dominio de σ
Límite	La suma formal de bordes en C	Bordes de X	Corresponde al límite de σ , que son los mapas inducidos sobre sus <i>faces</i>

Figura B.3: Características de las homologías simplicial, celular y singular.

B.0.4. Homología simplicial

Definición B.0.12. Sea X un simplicial complex. La **homología simplicial de X** , denotada por $H_*(X)$, es una secuencia de espacios vectoriales $H_k(X), k \in \mathbb{N}$ llamada “ k -dimensional homología de X ”.

Definición B.0.13 (Dimensión de $H_k(X)$ /clase o grupos homólogos/ *Betti-number*). La dimensión k , llamado el k -ésimo número *Betti* de X , es una medida basta del número de diferentes huecos en el espacio X que pueden ser percibidos mediante el uso de subcomplexes de dimensión k .

La construcción principal de esta homología es una versión algebraica de un simplicial complex, mediante la descomposición de un espacio en términos de piezas simples en una colección de espacios vectoriales (o módulos) y transformaciones lineales (u homomorfismos).

Además, por la aplicación de cadenas complexes C_k , se tienen las siguientes definiciones:

- Considerar los siguientes subespacios de C_k : los **ciclos** $Z_k(X)$ (*cycles* o subcomplexes sin frontera) y las **fronteras** $B_k(X)$ (k -boundaries o subcomplexes k -chains de mayor dimensión):

$$k - \text{cycles} : Z_k(X) = \ker(\partial : C_k \mapsto C_{k-1}) \quad (\text{B.6})$$

$$k - \text{boundaries} : B_k(X) = \text{img}(\partial : C_{k+1} \mapsto C_k) \quad (\text{B.7})$$

Así, B_k es un subespacio de Z_k , lo cual tiene las siguientes implicaciones:

- Los k -cycles $\in X$ son los objetos básicos que cuantifican un *hueco de dimensión k* en X .
- Además, es posible que algunos de los k -cycles $\in X$ estén midiendo al mismo hueco, incluso que otros ciclos no detecten un hueco del todo; por lo que estos delimitan un subcomplex de dimensión $(k + 1) \in X$.
- Se dice que dos ciclos $\{\xi, \eta\} \in Z_k(X)$ son *homólogos* si su diferencia es una frontera:

$$[\xi] = [\eta] \leftrightarrow \xi - \eta \in B_k(X) \quad (\text{B.8})$$

Observación B.0.14. Así, en lo que respecta a la medición de agujeros, los ciclos homólogos son equivalentes.

- Por lo anterior, la **homología k -dimensional de X** es el cociente de espacios vectoriales expresado en la ecuación B.9, es decir, el espacio vectorial apropiado para distinguir los ciclos homólogos:

$$H_k(X) = \frac{Z_k(X)}{B_k(X)} \quad (\text{B.9})$$

Observación B.0.15. El Lema B.0.9, implica que $B_k(X)$ es un subespacio de $Z_k(X)$, y por lo tanto, crea un espacio vectorial bien definido.

Observación B.0.16. En términos generales, al construir la homología, se está eliminando los ciclos que son fronteras de un subcomplex de orden superior del conjunto de todos los ciclos, para que los restantes lleven información sobre los huecos dimensionales del complex. Es decir, que cualquier elemento de $K_k(X)$ es una clase de equivalencia de ciclos k -cycles homólogos.

Por lo tanto, cada clase de homología no trivial en una determinada dimensión identifica un "hueco" correspondiente en esa dimensión. Como ejemplo de la realización geométrica de homologías, se listan los siguientes casos, considerando a X como un grafo topológico:

- $H_0(X)$.- Su base más simple consiste de una elección de vértices en X . La dimensión de esta homología es igual al número de componentes conectados de X , es decir, los tipos de huecos en X que los puntos pueden detectar.

Como $\partial_0 = 0$, todas las 0 -chains son 0 -cycles : $Z_0 = C_0$. Así, B_0 consiste de uniones finitas de vértices, con un número par en cada componente conectado de X . Al elegir el conjunto de vértices de X como una base para Z_0 , se tiene que cualquier par de elementos de la base son homólogos si y solo si ellos están en el mismo componente de ruta conectado de X . Así, la dimensión de $H_0(X)$ es el número de tales componentes de ruta conectados.

- $H_1(X)$.- Su base más simple consiste de una elección de lazos en X , cada uno de los cuales rodea un hueco diferente en X . Es decir, los elementos de $H_1(k)$ son colecciones finitas de ciclos orientados en el 1 -skeleton de X . Por lo tanto, $H_1(k)$ es una medida del número y tipos de ciclos.

k-Laplacianos combinatorios

Los grupos homológicos se aplican para distinguir espacios topológicos, mediante la identificación de huecos (de varias dimensiones) contenidos en tales espacios. A partir de que los operadores de frontera ∂_k tienen una representación matricial B_k , es posible definir su operador dual de la siguiente forma:

$$\partial_k^* : C_{k+1}(X) \mapsto C_k(X) \quad (\text{B.10})$$

A partir de la ecuación B.10, se define al operador k-Laplaciano discreto del simplicial complex de la siguiente forma:

$$\mathcal{L}_k : C_k(X) \mapsto C_k(X) = \partial_{k+1} \partial_{k+1}^* + \partial_k \partial_k^* \quad (\text{B.11})$$

Observación B.0.17. En [19], Egerstedt denota al operador Laplaciano como Δ_k , el cual permanece invariante en cada uno de sus sub espacios, además de ser positivo definido sobre las imágenes de ∂_{k+1} y ∂_k^* .

Este operador B.11 fue introducido en 1945 por Eckmann, quien notó que cada $C_k(X)$ puede descomponerse en subespacios ortogonales de la forma:

$$C_k(X) = \mathcal{H}_k(X) \oplus im(\partial_{k+1}) \oplus im(\partial_k^*) \quad (\text{B.12})$$

Donde:

$$\mathcal{H}_k(X) = \{c \in C_k(X) : \mathcal{L}_k c = 0\} = ker(\mathcal{L}_k) \quad (\text{B.13})$$

Así, de la ecuación B.13, se dan las siguientes observaciones:

Observación B.0.18. Para cada k , se tiene el siguiente isomorfismo:

$$H_k(X) \cong \mathcal{H}_k(X) \quad (\text{B.14})$$

Observación B.0.19. “Para calcular los grupos homológicos de un simplicial complex, es suficiente estudiar el espacio nulo de la matriz \mathcal{L}_i .”

Observación B.0.20. Por las observaciones dadas, se concluyen lo siguientes cálculos inmediatos:

- Al aplicar la representación matricial de operadores de frontera, se define al k -Laplaciano combinatorio de la siguiente forma:

$$L_k = B_k^T B_k + B_{k+1}^T B_{k+1} \in \mathbb{R}^{n_k \times n_k} \quad (\text{B.15})$$

donde B_k^T es la representación matricial de ∂_k y n_k es el número de k -simplices de X .

- El grafo Laplaciano común es:

$$\mathcal{L}_0 : C_0(X) \mapsto C_0(X) \quad (\text{B.16})$$

, es decir, L_0 es la matriz Laplaciana usual.

- En términos matriciales de adyacencia y de grado, los k -Laplacianos combinatorios se calculan de la siguiente forma:

$$L_k = D_U^{(k)} - A_U^{(k)} + (k+1)\mathbb{I}_{n_k} + A_L^{(k)} \quad (\text{B.17})$$

, donde $A_U^{(k)}$ y $A_L^{(k)}$ son las matrices de adyacencia superior e inferior, respectivamente, y $D_U^{(k)}$ es la matriz de grado superior.

Observación B.0.21. La representación expresada en la ecuación B.17 implica que la i -ésima fila de L_k depende únicamente de las interacciones locales entre el i -ésimo k -simplex con sus k -simplices adyacentes superiores e inferiores.

- Ya que no tienen sentido los simplices de dimensión negativa: $C_1(X) = 0$.
- Además, los mapas ∂_0 y ∂_0^* se asumen como mapas cero, por lo que $\mathcal{L}_0 = \partial_1 \partial_1^*$, donde ∂_1 es la matriz de incidencia del grafo.

B.0.5. Homología relativa

Por la forma de modelar al SDM en esta tesis, se requirió calcular el módulo de huecos en cierta región del espacio distribuido por nodos, por ejemplo, el considerar los nodos frontera del cúmulo general. Por esto, el concepto de homología relativa, así como los elementos que la conforman, son definidos a continuación.

Aunque comúnmente se toma a C_k a ser un módulo generado por k -dimensional - simplices, es posible también modificar C_k para dar lugar a homología reducida relativa a un subcomplex. Se parte de una $Y \subset X$ (subcomplex de X). Entonces, X, Y definen cadenas simplicial complex, cuyo resultado $C_k(Y) \subset C_k(X)$ es un submódulo para toda k .

Definición B.0.22. Se define como cadenas relativas:

$$C_k(X, Y) := \frac{C_k(X)}{C_k(Y)} \quad (\text{B.18})$$

al espacio cociente obtenido de $C_k(X)$ al “colapsar” el subespacio generado por los k -simplices en Y . Así, este cociente es estimado por ∂ , y los subespacios definidos por el kernel y la imagen satisfacen:

$$B_k(X, Y) \subset Z_k(X, Y) \subset C_k(X, Y) \quad (\text{B.19})$$

Observación B.0.23. De la definición B.0.22, el mapa frontera se extiende como:

$$\partial_k : C_k(X, Y) \mapsto C_{k-1}(X, Y) \quad (\text{B.20})$$

de forma que se preserve la condición $\partial^2 = 0$.

Definición B.0.24. Partiendo de las cadenas relativas, se define a la **homología relativa** como:

$$H_k(X, Y) = \frac{Z_k(X, Y)}{B_k(X, Y)} \quad (\text{B.21})$$

Observación B.0.25. Esta homología mide huecos detectados por las cadenas cuyas fronteras se encuentran en Y .

B.0.6. Mapas entre categorías: functorialidad

Definición B.0.26. Un **mapa cadena** $\varphi : C \mapsto C'$ es un homomorfismo sobre cadenas, respetando el grado y conmutación con el mapa frontera. Esto es mejor expresado en el siguiente *diagrama conmutativo*:

$$\begin{array}{ccccccccc} \cdots & \longrightarrow & C_{n+1} & \xrightarrow{\partial} & C_n & \longrightarrow & C_{n-1} & \longrightarrow & \cdots \\ \downarrow \varphi & & \downarrow \varphi & & \downarrow \varphi & & \downarrow \varphi & & \downarrow \varphi \\ \cdots & \longrightarrow & C'_{n+1} & \xrightarrow{\partial'} & C'_n & \xrightarrow{\partial'} & C'_{n-1} & \longrightarrow & \cdots \end{array}$$

Conmutación significa que el homomorfismo es trayectoria-independiente en el diagrama; ej., $\varphi \circ \partial = \partial' \circ \varphi$.

Un mapa cadena φ induce homomorfismos sobre grupos homológicos: $H(\varphi) : H(C) \mapsto H(C')$, enviando $[\zeta] \in H_k(C)$ hacia $[\varphi(\zeta)] \in H_k(C')$. Así, $H(\varphi)$ es la secuencia completa de **homomorfismos inducidos** sobre la homología; por lo que la homología es **functorial**, es decir, que los homomorfismos inducidos respetan la composición de mapas cadena.

B.0.7. Complexes en homología

Vietoris-Rips complexes

Se considera un subconjunto discreto $Q \in \mathbb{R}^n$. Al complex Q se le asocia como una colección de puntos de datos (*nube de puntos*) muestreados de una subvariedad, por lo que es deseable reconstruir tal subvariedad fundamental de la nube de puntos.

Además, al elegir un parámetro de longitud $\epsilon > 0$, el *Vietoris-Rips complex* de escala ϵ sobre Q , denotado por $\mathcal{VR}_\epsilon(Q)$, corresponde al simplicial complex cuyos simplices son todas aquellas colecciones finitas de puntos en Q con distancia entre pares de a lo sumo ϵ (cerrado sobre subconjuntos). Así, $\mathcal{VR}_\epsilon(Q)$ da una buena aproximación a la estructura fundamental de la nube de puntos mediante la correcta elección de ϵ .

El grado con el que $\mathcal{VR}_\epsilon(Q)$ (para un ϵ fijo) captura la topología de la nube de puntos no es obvio. Por ejemplo, al considerar el mapa de proyección $\delta : \mathcal{VR}_\epsilon(Q) \mapsto \mathbb{R}^n$, el cual toma los vértices $\mathcal{VR}_\epsilon(Q)^0$ hacia Q , así como los k -*simplex* de $\mathcal{VR}_\epsilon(Q)$ hacia el casco convexo de los vértices asociados en Q . Así, la imagen de δ en \mathbb{R}^n es llamada *sombra* Sh del $\mathcal{VR}_\epsilon(Q)$. Por lo regular, $\mathcal{VR}_\epsilon(Q)$ y Sh son no homeomórficos, debido a que el dominio de proyección δ es probable que tenga mayor dimensión que el codominio.

Čech complexes y muestreos aleatorios

El $\mathcal{VR}_\epsilon(Q)$ es un ejemplo de una clase más general de simplicial complexes que logran rellenar una estructura determinada. Para codificar un simplicial complex típico en una forma computacional que el software pueda manejar, es necesario una lista completa de todos los simplices.

Dado una nube de puntos $Q \in \mathbb{R}^n$ y un parámetro de longitud $\epsilon > 0$, el *Čech complex* \check{C}_ϵ a ser el simplicial complex construido sobre Q se define como sigue:

Definición B.0.27. El *Čech complex* de Q , denotado por $\check{C}_\epsilon(Q)$, es la colección de $k + 1$ distintos elementos (x_i 's) de Q tal que la red de intersecciones de bolas de diámetro ϵ en los x_i 's es no vacía. Además: $\check{C}_\epsilon(Q) \subset \mathcal{VR}_\epsilon(Q)$.

Sin embargo, en los $\check{C}_\epsilon(Q)$ no es posible solamente almacenar el $1 - skeleton$ y rellenar el resto, ya que se deben revisar muchas intersecciones para construir el complex completo.

B.0.8. Homología persistente

Se trata de un método algebraico para discernir características topológicas (ej. componentes, agujeros, estructura gráfica) de datos (ej. conjunto de puntos discretos con métrica bien definida entre pares).

Bibliografía

- [1] *A decentralized algorithm for spectral analysis*. página 14, 2007, ISSN 07349025.
- [2] Arana, Oscar: *Estudio y desarrollo de técnicas de privacidad geográfica en redes inalámbricas*. phdthesis, Posgrado en Ingeniería Eléctrica (Telecomunicaciones). UNAM, Ciudad de México, Enero 2019.
- [3] Benítez-Pérez, Héctor, Jorge L. Ortega-Arjona, Paul E. Méndez-Monroy, Ernesto Rubio-Acosta y Oscar A. Esquivel-Flores: *Control Strategies and Co-Design of Networked Control Systems*, volumen 13. Springer Nature, Mexico City, 1^a edición, 2019, ISBN 978-3-319-97042-4. <http://link.springer.com/10.1007/978-3-319-97044-8>.
- [4] Castañeda, Armando, Sergio Rajsbaum y Matthieu Roy: *Convergence and covering on graphs for wait-free robots*. Journal of the Brazilian Computer Society, 24(1):1–15, 2018, ISSN 16784804.
- [5] Chintakunta, Harish y Hamid Krim: *Topological Fidelity in Sensor Networks*. páginas 4–5, 2011. <http://arxiv.org/abs/1106.6069>.
- [6] Crawford, Joseph y Tijana Milenković: *ClueNet: Clustering a temporal network based on topological similarity rather than denseness*. PLoS ONE, 13(5), may 2018, ISSN 19326203.
- [7] de Silva, Vin y Robert Ghrist: *Coverage in sensor networks via persistent homology*. Robotics: Science and Systems, 2005. <https://www.math.upenn.edu/~ghrist/preprints/persistent.pdf>.
- [8] Dłotko, P., Ghrist R. Juda M.: *Distributed computation of coverage in sensor networks by homological methods*. En *Applicable Algebra in Engineering, Communication and Computing* [8], páginas 29–58. <https://doi.org/10.1007/s00200-012-0167-7>.
- [9] Gafni, Eli, Petr Kuznetsov y Ciprian Manolescu: *A generalized asynchronous computability theorem*. ACM, página 10, Julio 2014. <https://dl.acm.org/citation.cfm?doid=2611462.2611477>.
- [10] Ghrist, Robert: *Homological algebra and data*. En *IAS/Park City Mathematics Series* [10], página 53. <https://www.math.upenn.edu/~ghrist/preprints/HAD.pdf>.
- [11] Ghrist, Robert: *Applied algebraic topology & sensor networks*. Notes for AMS short course, 1^a edición, 2010. Estas notas contienen resultados con Raryshnikov, Vin de Silva, y Michael Robinson.
- [12] Ghrist, Robert: *Elementary applied topology*. Mathematics and Electrical & Systems Engineering. University of Pennsylvania, 2014.

- [13] Herlihy, Maurice; Kozlov, Dmitry; Rajsbaum Sergio;: *Chapter 1 Introduction*, 2014. <http://cs.brown.edu/courses/cs2951s/Ch01.pdf>.
- [14] Herlihy, Maurice; Kozlov, Dmitry; Rajsbaum Sergio;: *Distributed computing through combinatorial topology*. Morgan Kaufmann, 2014.
- [15] Herlihy, Maurice y Sergio Rajsbaum: *Algebraic topology and distributed computing a primer*. Computer Science Today. Lecture Notes in Computer Science, 1000:14, 1995.
- [16] Herlihy, Maurice; Shavit, Nir: *The topological structure of asynchronous computability*. ACM, 1999.
- [17] Kempe, David, Jon Kleinberg y Amit Kumar: *Connectivity and inference problems for temporal networks*. Journal of Computer and System Sciences, 64(4):820–842, 2002, ISSN 10902724.
- [18] Medel Juárez, José de Jesús; Guevara López, Pedro y Daniel Cruz Pérez: *Temas Selectos de Sistemas en Tiempo Real*, volumen 1. Instituto Politécnico Nacional, Ciudad de México, first edición, 2007.
- [19] Mesbahi, Mehran y Magnus Egerstedt: *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, New Jersey, 2010, ISBN 9780691140612.
- [20] Mohamed, Shaimaa M, Haitham S Hamza y Iman Aly Saroit: *Introduction US CR Coverage in Mobile Wireless Sensor Networks*. Computer Communications, 110:133–150, 2017, ISSN 0140-3664. <http://dx.doi.org/10.1016/j.comcom.2017.06.010>.
- [21] Muhammad, Abubakr y Ali Jadbabaie: *Decentralized computation of homology groups in networks by gossip*. Proceedings of the American Control Conference, páginas 3438–3443, 2007, ISSN 07431619.
- [22] Munkres, James R.: *Elements of algebraic topology*. CRC Press, 2018.
- [23] Ni, Chien Chun, Zhengyu Su, Jie Gao y Xianfeng David Gu: *Capacitated kinetic clustering in mobile networks by optimal transportation theory*. Proceedings - IEEE INFOCOM, 2016-July, 2016, ISSN 0743166X.
- [24] Silva, Vin de y Robert Ghrist: *Coordinate-free coverage in sensor networks with controlled boundaries via homology*. International Journal of Robotics Research, 25:30, Diciembre 2007.
- [25] Silva, Vin de y Robert Ghrist: *Homological sensor networks*. Notices of the AMS (American Mathematical Society), 54(1):10–17, Enero 2007.
- [26] Wang, Tao y William N. N. Hung: *Reliable Node Clustering for Mobile Ad Hoc Networks*. Journal of Applied Mathematics, 2013(iii):1–8, 2013, ISSN 1110-757X.
- [27] Yang, Hua y Zhi mei Li: *A Genetic-algorithm-based Weighted Clustering Algorithm in MANET*. International Journal of Future Generation Communication and Networking, 10(3):31–40, 2017, ISSN 22337857.