

Descomposición Óptima en Subframeworks con Aplicaciones al Control de Redes Multirobot

Juan F. Presenza^{1,3}, Ignacio Mas^{2,3}, Juan I. Giribet^{2,3}, J. Ignacio Alvarez-Hamelin^{1,3}

¹ Facultad de Ingeniería, Universidad de Buenos Aires

² Laboratorio de Inteligencia Artificial y Robótica - Universidad de San Andrés, Buenos Aires, Argentina.

³ CONICET - Argentina

Resumen—Este trabajo presenta una extensión del formalismo que captura la de rigidez de un framework a partir de la subdivisión del mismo en subframeworks. Este abordaje permite modelar una red de robots móviles, y ha demostrado su utilidad en el control de redes multirobot permitiendo el diseño de controladores descentralizados y escalables. Teniendo en cuenta métricas realistas del control multirobot como el tiempo de retardo y la carga de las comunicaciones entre agentes, este trabajo compara cualitativa y cuantitativamente distintos tipos de descomposición. A partir de este análisis, se establecen criterios de optimalidad que permiten seleccionar descomposiciones óptimas. A su vez, se presenta un algoritmo escalable que permite aproximar estas soluciones óptimas. Mediante simulaciones computacionales, se valida el esquema propuesto y se analizan los resultados.

I. INTRODUCCIÓN

En la última década, la coordinación de sistemas de múltiples robots móviles interconectados se ha convertido en un área de intensa investigación, motivada por las ventajas en adaptabilidad, robustez y escalabilidad en una amplia variedad de aplicaciones [1], [2]. La utilización de esquemas de control centralizados, donde un centro de cómputo debe procesar todas las variables del sistema, no resulta apropiada cuando la cantidad de robots es elevada. Esto se debe a principalmente a los altos requerimientos de capacidad computacional del nodo central, los efectos de cuello de botella en los canales de comunicación, y falta de robustez por la existencia de un punto de falla. Todas ellas se traducen en una baja escalabilidad. Por el contrario, en las técnicas descentralizadas, las comunicaciones, el cómputo y la decisión se reparten entre los agentes. Los mismos actúan únicamente en función de la percepción de su entorno a través de sensores, y de las interacciones con otros vehículos a través de comunicaciones punto-a-punto. Se logra así dotar a los esquemas de una mayor escalabilidad, a expensas de una mayor dificultad para que los agentes accedan a la información asociada a la red.

En particular, numerosos desafíos se presentan cuando los agentes únicamente cuentan con sensores que les permiten acceder a información relativa a otros robots, por ejemplo, posición relativa o distancia. En tales casos, la topología de la red subyacente, dada por las interacciones entre los agentes, tiene una gran importancia. Más aún, esta topología suele ser dinámica, es decir, permite la creación y destrucción de enlaces, producto del movimiento de los robots o de cambios en el ambiente circundante. Un caso de particular

importancia en la literatura se presenta cuando los robots cuentan únicamente con mediciones de distancia respecto de sus vecinos [3], [4]. La rigidez de los grafos resulta ser la propiedad necesaria y suficiente para controlar al conjunto de robots a partir de dichas mediciones. Por este motivo, se han realizado importantes esfuerzos para diseñar estrategias de control que permitan mantener o recuperar la rigidez en redes con topología dinámica.

Las estrategias clásicas de control de rigidez pueden ser clasificadas como continuas [5], [6], [7] y combinatorias [8], [9]. En los esquemas continuos, el controlador guía el movimiento de los robots de forma tal de mantener la rigidez del grafo, permitiendo cambios en la topología de la red. Esto se logra a través de la maximización del autovalor de rigidez, un parámetro que sirve como medida del grado de rigidez de una grafo. La implementación descentralizada se basa en la utilización de filtros de consenso que permiten obtener localmente una estimación del autovalor de rigidez y del correspondiente autovector, necesarios para computar la acción de control. El desempeño del control depende en gran medida de la capacidad de estimar rápidamente estos parámetros. Sin embargo, la velocidad de convergencia de los estimadores está relacionada inversamente con el diámetro de la red [10]. Además, cuando el número de conexiones mantenidas por cada robot es limitado, el diámetro crece al menos logarítmicamente con el número de robots, imponiendo un claro límite a la escalabilidad de estos abordajes.

Alternativamente, en los enfoques combinatorios se diseñan reglas locales para la adición y remoción de enlaces entre agentes, las cuales son luego empleadas para dictar los movimientos de los robots. La naturaleza discreta de estos esquemas los dotan de mayor eficiencia, dado que únicamente requieren realizar cálculos e intercambio de información durante las transiciones, es decir, ante la inminente pérdida o creación de un enlace. Estos esquemas poseen mejor escalabilidad dado que las acciones de control se toman con datos locales, sin requerir información global. No obstante, son conservativos en cuanto a la eliminación de enlaces, debido a que conexiones redundantes pueden no ser detectadas de forma local. Al ser enfoques locales, no se puede emplear el autovalor de rigidez ni ninguna medida de rigidez que permite definir un criterio de optimalidad relacionado.

En [11], [12] se buscó superar estas limitaciones a través

de la definición de subframeworks. A partir del análisis de rigidez de los mismos, se puede determinar la rigidez de todo el framework. Esto elimina la necesidad de intercambiar información de extremo a extremo de la red, lo que mejora notablemente la escalabilidad del sistema. De esta manera, se diseñó un controlador descentralizado capaz maximizar los autovalores de rigidez de los subframeworks, y de esta manera garantizando la rigidez de todo el framework. Para el funcionamiento del controlador, la información debe ser propagada en el interior de cada subframework, por lo que el tamaño de los mismos influye significativamente en el desempeño del control. El propósito de este trabajo es analizar este comportamiento y establecer un criterio de optimalidad para la elección de los subframeworks que reflejen objetivos realistas de un sistema de control multirobot.

II. LA RIGIDEZ DE LOS FRAMEWORKS

II-A. Preliminares

Modelamos una red multirobot mediante un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ donde $\mathcal{V} = \{1, \dots, n\}$ representa el conjunto de agentes y $\mathcal{E} \subseteq \{\{i, j\} \mid i, j \in \mathcal{V}, i \neq j\}$ el conjunto de m enlaces de comunicación punto-a-punto. El framework $\mathcal{F} = (\mathcal{G}, p)$ consiste en una realización de \mathcal{G} dada por la función $p : \mathcal{V} \rightarrow \mathbb{R}^d$ que asigna posiciones, a los nodos, en un espacio d -dimensional. Decimos que un framework es completo si tiene todos los enlaces posibles, es decir, $m = n(n-1)/2$. El número de conexiones de un nodo se representa mediante el grado del vértice $\delta(i)$. La distancia geodésica $d(i, j)$ entre dos nodos i, j es la mínima cantidad de enlaces que deben atravesarse para conectarlos; la excentricidad de un vértice $e(i) = \max_j d(i, j)$, y el diámetro del grafo $\text{diam}(\mathcal{G}) = \max_i e(i)$.

II-B. Rigidez de Frameworks

Intuitivamente, decimos que un framework (\mathcal{G}, p) es rígido cuando, conocido el conjunto de distancias euclídeas asociadas a los enlaces $\{z_{ij} \mid \{i, j\} \in \mathcal{E}\}$, es posible reconstruir el conjunto de posiciones $\{p(i) \mid i \in \mathcal{V}\}$, excepto por rotaciones, reflexiones y traslaciones. A continuación, se presenta la definición formal de este concepto [13].

Dos frameworks (\mathcal{G}, p) y (\mathcal{G}, q) se dicen equivalentes si se cumple

$$\|p(i) - p(j)\| = \|q(i) - q(j)\| \quad \text{para todo } \{i, j\} \in \mathcal{E}; \quad (1)$$

y se dicen congruentes si

$$\|p(i) - p(j)\| = \|q(i) - q(j)\| \quad \text{para todo } i \in \mathcal{V}, j \in \mathcal{V}. \quad (2)$$

El framework (\mathcal{G}, p) es rígido si existe $\varepsilon > 0$ tal que para toda realización q que satisface $\|p(i) - q(i)\| < \varepsilon$ para todo i , se cumple que (1) implica (2). Caso contrario, se dice que (\mathcal{G}, p) es flexible.

Para establecer la rigidez mediante métodos computacionales, primero definimos la matriz de rigidez $R \in \mathbb{R}^{m \times dn}$ de un framework como el jacobiano de la función

$$f(\mathbf{p}) = [\dots \|p(i) - p(j)\| \dots]^T, \quad \mathbf{p} = [p(1)^T \dots p(n)^T]^T$$

donde cada componente del vector $f(\mathbf{p})$ corresponde a un enlace. Si ordenamos los autovalores $\lambda_1, \dots, \lambda_{dn}$ de $R^T R$ de forma creciente, entonces se puede probar (ver [5]) que un framework es rígido si y solo si

$$\lambda_{\frac{d(d+1)}{2}+1}(R^T R) > 0. \quad (3)$$

Dicho valor es denominado autovalor de rigidez del framework, y sirve no solo como condición necesaria y suficiente, sino también como una medida de su grado de rigidez.

III. DESCOMPOSICIÓN EN SUBFRAMEWORKS

En [11], [12] se presentó un esquema de descomposición de frameworks que fue utilizado para el control descentralizado de rigidez. El mismo se define a continuación.

Definición 1 (Descomposición densa). *Sea \mathcal{F} un framework y $h : \mathcal{V} \rightarrow \mathbb{Z}_{>0}$ una función que asocia un alcance (medido en distancia geodésica) a cada nodo. Se llama descomposición densa al conjunto $\text{dense}(\mathcal{F}) = \{\mathcal{F}_1, \dots, \mathcal{F}_n\}$ donde $\mathcal{F}_i = (\mathcal{G}_i, p)$ tal que*

$$\mathcal{V}_i = \{j \mid d(i, j) \leq h(i)\}, \mathcal{E}_i = \{\{j, k\} \in \mathcal{E} \mid j, k \in \mathcal{V}_i\}. \quad (4)$$

Al robot i se le llama centro del subframework \mathcal{F}_i , y a $h(i)$ se le dice alcance del mismo.

Notar que, para un mismo framework, pueden existir múltiples descomposiciones, correspondientes a las distintas funciones de alcance h . Si $h(i) = 0$ decimos que el subframework \mathcal{F}_i es trivial, y definimos a la descomposición trivial como aquella donde todo subframework es trivial. Análogamente, un subframework es unitario si $h(i) = 1$, y la descomposición unitaria es aquella donde esto ocurre para todo i . Por otro lado, decimos que una descomposición es impropia si existe algún subframework \mathcal{F}_i que cubre todo el framework, es decir, $h(i) = e(i)$.

La rigidez de \mathcal{F} queda determinada por el siguiente teorema, el cual permite expresarla como un conjunto de condiciones locales.

Teorema 1 (Rigidez por Descomposición Densa [11]). *Sea \mathcal{G} conexo. Entonces \mathcal{F} es rígido si y sólo si existe una descomposición $\text{dense}(\mathcal{F})$ tal que todos los subframeworks son rígidos.*

Notar que este teorema, si bien indica las condiciones, no especifica cómo debe realizarse la descomposición; en otras palabras, resta analizar cómo elegir los alcances $h(1), \dots, h(n)$ de forma apropiada. Como se verá en el apartado III-B, cuando la descomposición es densa, la elección de un vector de alcances óptimo resulta directa. En cambio, cuando la descomposición no es densa, la elección no es tan simple. Este caso es analizado en profundidad en la Sección IV.

III-A. Métricas de la descomposición

Para comparar de forma cuantitativa las distintas descomposiciones, es necesario definir métricas sobre las mismas, que permitan evaluar algún comportamiento deseado.

En primer lugar, para implementar un algoritmo de rigidez de forma descentralizada, los agentes deben intercambiar paquetes de información que pueden recorrer caminos de distinta longitud hasta llegar a su destino. Por lo tanto, es necesario considerar el tiempo transcurrido durante ese recorrido. En los esquemas de control por subframeworks, los nodos de un subframework i se comunican entre sí enrutando mensajes a través del centro, resultando en caminos de longitud menor o igual al doble del alcance $h(i)$. Esto se observa en la Fig. 1a. Por ello, definimos el máximo retardo asociado a una descomposición como:

$$\mathcal{D}(h) = 2 \max_{i \in \mathcal{V}} h(i). \quad (5)$$

Notar que $2 \leq \mathcal{D}(h) \leq 2 \text{diam}(\mathcal{G})$, extremos que se alcanzan cuando la descomposición es unitaria, y cuando es impropia, respectivamente. Empleamos $\mathcal{D}_{\text{ref}}(\mathcal{G}) = \text{diam}(\mathcal{G})$ como parámetro de referencia para esta métrica, ya que es el mínimo retardo asociado a los algoritmos de intercambio de información global, como los protocolos de consenso.

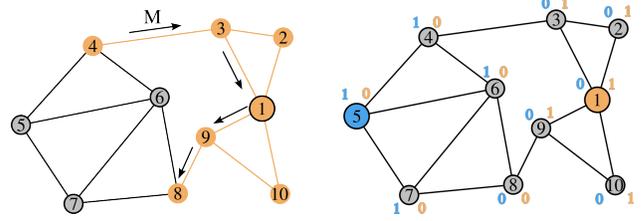
En segundo lugar, el intercambio de información bidireccional entre los nodos de un subframework y su centro, implica la utilización de los canales de comunicación. El esquema de control presentado en [12], requiere que cada robot emita dos tokens: T_A con información de las acciones de control, y T_S con datos del estado del mismo; los cuales son propagados por la red hasta llegar a sus respectivos destinos. El token acción emitido por i debe llegar a todos los nodos del subframework \mathcal{F}_i , por lo que es enviado a $h(i)$ hops de distancia; mientras que el de estado es enviado a todo robot cuyo subframework contiene a i , es decir a los nodos $\{j \mid d(i, j) \leq h(j)\}$. Esto implica que T_S debe ser enviado a $s(i)$ hops, donde $s(i) = \max\{h(j) \mid d(i, j) \leq h(j)\}$.

De esta forma, a medida que el alcance de los subframeworks crece, los tokens deben viajar una mayor distancia, lo que produce un aumento de la carga sobre los enlaces. Dada la descomposición dictada por el alcance h , para cuantificar la carga total de comunicación asociada a la propagación de tokens T_A, T_S , proponemos la función

$$\mathcal{L}(h) = \sum_{i \in \mathcal{V}} (N_A(i) + N_S(i)) \delta(i), \quad (6)$$

donde $N_A(i) = |\{j \mid d(i, j) < h(j)\}|$ y $N_S(i) = |\{j \mid d(i, j) < s(j)\}|$ representan la cantidad de tokens de acción y de estado que pasan por el robot i , respectivamente. Esto se observa en Fig. 1b. \mathcal{L} mide la cantidad total de veces que un enlace es empleado para transmitir un token. El valor $\mathcal{L}_{\text{ref}}(\mathcal{G}) = 2m$, donde m es el número total de enlaces, es tomado como referencia, dado que representa la utilización de los canales en una regla de vecinos más cercanos.

En [11] se empleó, de forma exitosa, la función \mathcal{L} como función de costo en un controlador de gradiente descendiente para guiar el movimiento de los robots. De esta forma, se logró inducir, en los robots, trayectorias que tienden a eliminar los enlaces con mayor carga, manteniendo la rigidez del sistema.



(a) Camino de un mensaje en \mathcal{F}_1 . (b) Carga impuesta por \mathcal{F}_1 y \mathcal{F}_5 .

Figura 1: Métricas para comparar distintas descomposiciones en subframeworks. En (a) se muestra el subframework \mathcal{F}_1 con $h(1) = 2$. En (b), se muestran las cargas impuestas por dos tokens: uno de \mathcal{F}_1 que debe viajar 2 hops y otro \mathcal{F}_5 que debe viajar 1 hop.

III-B. Alcance mínimo de rigidez

Puede observarse que \mathcal{D}, \mathcal{L} son funciones no decrecientes respecto a los valores $h(1), \dots, h(n)$. De esta forma, para minimizar ambas métricas, es apropiado seleccionar los menores alcances posibles. Entonces, si $\mathcal{F}_i[\eta]$ denota el subframework con centro en i y alcance η , definimos el alcance mínimo de rigidez como

$$h_R(i) = \min\{\eta \mid \eta > 0 \text{ y } \mathcal{F}_i[\eta] \text{ es rígido}\}. \quad (7)$$

Luego, h_R es la descomposición que minimiza \mathcal{D} y \mathcal{L} cuando la descomposición es densa, y por lo tanto es la elegida en dichos casos. Es interesante notar que, desde el punto de vista del intercambio de información, $\max_i h_R(i)$ es una medida del grado de descentralización lograda por la descomposición densa. Esto es, si h_R es unitaria, entonces es posible implementar un control de rigidez mediante interacciones entre vecinos únicamente, lo cual representa el mayor grado de descentralización posible. Por otro lado, si h_R es impropia, la descomposición en subframeworks no aporta ventajas respecto de un esquema centralizado.

IV. DESCOMPOSICIÓN DISPERSA

La descomposición dispersa es una modificación del esquema presentado anteriormente, la cual permite extender la familia de descomposiciones válidas. Antes de su presentación formal, el siguiente análisis permitirá comprender la necesidad de su definición.

Dado el grafo rígido de Fig. 2a, la descomposición densa óptima está dada por $h_R(1) = h_R(3) = h_R(4) = 3$ y $h_R(\cdot) = 1$ para el resto. Se puede observar que $\mathcal{F}_1 = \mathcal{F}_3 = \mathcal{F}_4 = \mathcal{F}$. De esta forma, analizar la rigidez de \mathcal{F} mediante la descomposición densa, no presenta ventajas respecto de un enfoque centralizado; pero además se incurre en una gran superposición de subframeworks debido a que, para todo i , $\mathcal{F}_i \subseteq \mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_4$. Esto se evidencia en los altos valores que toman las métricas: $\mathcal{D}(h_R) = 1.5 \cdot \mathcal{D}_{\text{ref}}$ y $\mathcal{L}(h_R) \approx 8.9 \cdot \mathcal{L}_{\text{ref}}(\mathcal{G})$.

Definición 2 (Descomposición dispersa). Sea \mathcal{F} un framework y $h : \mathcal{V} \rightarrow \mathbb{Z}_{\geq 0}$ una función que asocia un alcance (medido en distancia geodésica) a cada nodo. Sea $\mathcal{F}_1, \dots, \mathcal{F}_n$ el conjunto de subframeworks definidos según (4), y $\mathcal{E}_L \subseteq \mathcal{E}$ el conjunto de

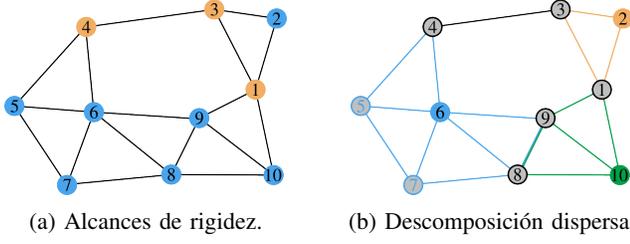


Figura 2: En (a), el color de los nodos según su alcance de rigidez h_R ; en (b) Descomposición dispersa dada por $h(2) = h(6) = h(10) = 1$.

enlaces que no pertenecen a ningún subframework. Entonces, se llama descomposición dispersa al conjunto

$$\text{disp}(\mathcal{F}) = \{\mathcal{F}_i \mid \exists j : \mathcal{F}_i \subseteq \mathcal{F}_j\} \cup \{\mathcal{L}_\ell \mid \ell \in \mathcal{E}_L\} \quad (8)$$

donde $\mathcal{L}_\ell = (\mathcal{G}_\ell, p)$ y $\mathcal{G}_\ell = (\ell, \{\ell\})$.

Cada descomposición dispersa queda determinada por una función de alcances h , sin embargo, dos funciones distintas podrían inducir la misma descomposición. Como se verá en el resto del trabajo, la descomposición dispersa permite obtener una reducción del número y del tamaño de los subframeworks empleados. De esta forma, se logra reducir el retardo máximo y la carga de comunicaciones, mejorando las características de la descomposición.

Retomando el ejemplo de la Fig. 2a, supongamos que se elige la descomposición dispersa dada por $h(1) = 3$ y $h(\cdot) = 0$ para el resto; es decir $\text{disp}(\mathcal{F}) = \{\mathcal{F}_1\}$. De esta forma se elimina la superposición existente, obteniéndose $\mathcal{D}(h) = 1.5 \cdot \mathcal{D}_{\text{ref}}$ y $\mathcal{L}(h) \approx 4.3 \cdot \mathcal{L}_{\text{ref}}(\mathcal{G})$. No obstante, el retardo máximo permanece constante, como consecuencia de resultar en un esquema centralizado en el nodo $i = 1$, lo cual tampoco es deseable. Además, la carga \mathcal{L} sigue teniendo un valor elevado, producto de que todos los robots (excepto el centro) deben enviar sus tokens de estado a este último, los cuales en algunos casos deben recorrer toda la red. En la Fig. 2 se muestra una descomposición del mismo framework con $h(2) = h(6) = h(10) = 1$, y $h(\cdot) = 0$ para el resto. En este caso, se obtiene $\text{disp}(\mathcal{F}) = \{\mathcal{F}_2, \mathcal{F}_6, \mathcal{F}_{10}, \mathcal{L}_{\{3,4\}}\}$, y las métricas $\mathcal{D}(h) = 0.5 \cdot \mathcal{D}_{\text{ref}}$ y $\mathcal{L}(h) = \mathcal{L}_{\text{ref}}$, es decir, se reducen sustancialmente ambas métricas, incluso alcanzando valores de referencia. Por estas razones, es apropiado analizar exhaustivamente el conjunto de descomposiciones posibles y seleccionar aquellas que minimicen dichas medidas.

IV-A. Rigidez

Para determinar la rigidez de un framework mediante una descomposición dispersa, es necesario reformular el Teorema 1. Para ello definimos la operación K , tal que $K(\mathcal{F})$ es el framework completo inducido por \mathcal{F} . Luego, definimos \mathcal{F}_K , una versión modificada de \mathcal{F} , que surge de reemplazar cada componente \mathcal{C} de $\text{disp}(\mathcal{F})$ por $K(\mathcal{C})$, es decir,

$$\mathcal{F}_K = \bigcup_{\mathcal{C} \in \text{disp}(\mathcal{F})} K(\mathcal{C}). \quad (9)$$

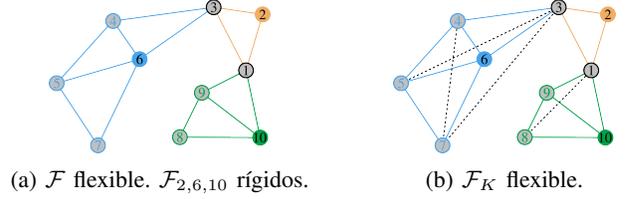


Figura 3: Los subframeworks 2, 6 y 10 son rígidos, pero no están rígidamente vinculados.

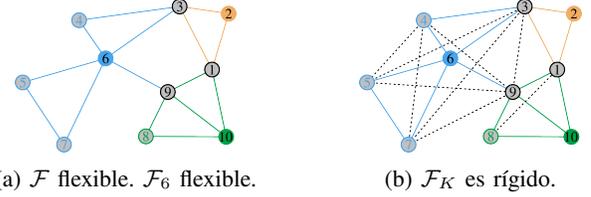


Figura 4: \mathcal{F} es flexible, pero los subframeworks están rígidamente vinculados. La flexibilidad proviene de \mathcal{F}_6 .

Definición 3 (Vinculación rígida). Sea una descomposición dispersa de \mathcal{F} . Decimos que el conjunto $\text{disp}(\mathcal{F})$ está rígidamente vinculado si y sólo si \mathcal{F}_K es rígido.

Esta definición es útil para separar la rigidez por subframeworks en dos condiciones: la primera referida la rigidez de los subframeworks, y la segunda a la forma en la que los mismos están vinculados. Es decir, si un framework es flexible pero todos sus subframeworks rígidos, entonces los mismos no pueden estar rígidamente vinculados. Esto se ilustra en Fig. 3 y Fig. 4. En la primera, se tiene \mathcal{F} flexible (a pesar de que $\mathcal{F}_2, \mathcal{F}_6, \mathcal{F}_{10}$ sean rígidos), y se observa que \mathcal{F}_K también es flexible; por lo que se concluye que los subframeworks no están rígidamente vinculados. En cambio, en la segunda, \mathcal{F} también es flexible, pero \mathcal{F}_K es rígido, es decir, los subframeworks están rígidamente vinculados, por lo que al menos un subframework debe ser flexible.

Teorema 2 (Rigidez por Descomposición Dispersa). \mathcal{F} es rígido si y solo si existe una descomposición $\text{disp}(\mathcal{F})$ tal que son todos sus componentes son rígidos y están rígidamente vinculados.

La prueba de este teorema puede encontrarse en [14, Capítulo 4]. Vale notar que, los componentes dados por los subframeworks triviales y por los enlaces vinculantes, son rígidos por definición. El siguiente lema, que es consecuencia del teorema anterior, es útil para el estudio de la descomposición óptima que presentamos en la siguiente subsección.

Lema 1. Si \mathcal{F} es rígido, entonces para toda descomposición, los componentes asociados están rígidamente vinculados.

IV-B. Descomposición Óptima

Como ya se presentó, es importante abordar el problema de elegir, dado un framework rígido, una descomposición en subframeworks óptima, teniendo en cuenta las métricas de desempeño \mathcal{D}, \mathcal{L} . Esta descomposición, para que permita

establecer la rigidez del framework, deberá cumplir las condiciones del Teorema 2. No obstante, gracias al Lema 1, para descomponer un framework rígido no es necesario evaluar la condición de vinculación rígida.

En el caso de descomposición densa, se vio que el alcance mínimo de rigidez (h_R) minimiza ambas funciones. Sin embargo, en el esquema disperso, el conjunto de funciones h posibles es estrictamente mayor al anterior. Surge naturalmente entonces el problema de seleccionar una función de alcance de forma óptima en este nuevo conjunto. En primer lugar, es deseable elegir h tal que el retardo máximo sea menor o igual que el diámetro del grafo, por lo que imponemos la condición $\mathcal{D}(h) \leq \mathcal{D}_{\text{ref}}$. En segundo lugar, queremos minimizar la carga comunicacional, por lo que es propicio emplear \mathcal{L} como un costo a minimizar.

Sin embargo, notamos que la descomposición trivial alcanza el mínimo global de ambas medidas. Esto no es útil desde el punto de vista del control, dado que las condiciones de rigidez por subframeworks del Teorema 2 (que son las que el controlador debe monitorear), equivalen a la rigidez de todo el framework. En otras palabras, los enlaces vinculantes presentan una dificultad para el controlador, ya que no hay subframework que los supervise y decida si se puede perder o no. Debe plantearse otra estrategia para tratar con los mismos; como caso más conservador, mantenerlos en todo momento.

Por este motivo, agregamos a la función de costo un término que incluye a la cantidad de enlaces vinculantes m_ℓ existentes en una descomposición, de forma tal de mantenerla acotada. Luego, definimos el alcance óptimo

$$h^* = \underset{h \in \mathcal{H}}{\text{argmín}} \mathcal{L}(h) + 2m_\ell, \quad (10)$$

donde el espacio de búsqueda es

$$\mathcal{H} = \{h \mid \forall \mathcal{C} \in \text{disp}(\mathcal{F}) : \mathcal{C} \text{ es rígido, y } \mathcal{D}(h) \leq \mathcal{D}_{\text{ref}}\}. \quad (11)$$

La búsqueda exhaustiva de h^* , no resulta un abordaje escalable con el número de robots n . Esto se debe a que debe evaluarse la rigidez de todos sus componentes para cada una de las $(\mathcal{D}_{\text{ref}} + 1)^n$ descomposiciones posibles.

Para solucionar esto, proponemos un algoritmo *greedy* que parte de h_R , y en cada iteración, lleva a cero el alcance del subframework que provoca el mayor decrecimiento de la función de costo. Llamamos \hat{h}^* a la descomposición subóptima obtenida. El procedimiento se muestra en el Algoritmo 1, que aproxima el valor de h^* buscando entre no más $n(n+1)/2$ alcances distintos.

V. RESULTADOS

En Fig. 5 y Fig. 6 se muestran dos ejemplos del alcance de rigidez obtenido y la aplicación del Algoritmo 1 para obtener el alcance subóptimo, y las métricas que lo acompañan. En ambos casos, puede observarse cómo la descomposición dispersa es capaz de capturar la rigidez del framework a partir de un conjunto reducido de componentes, eliminando aquellos de

Algoritmo 1: Búsqueda de descomposición óptima.

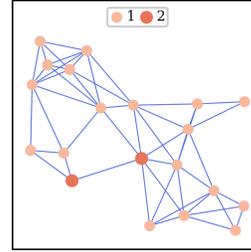
Input: \mathcal{F} (framework rígido), h_R (alcance mínimo de rigidez)

Output: \hat{h}^* (alcance subóptimo)

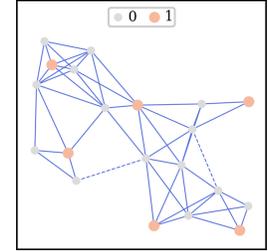
```

 $\hat{h}^* \leftarrow h_R$ ;
remain  $\leftarrow \{1, \dots, n\}$ ;
terminar  $\leftarrow \text{False}$ ;
mín  $\leftarrow \infty$ ;
while not terminar do
  collapse  $\leftarrow \text{None}$ ;
  for  $i$  in remain do
     $g \leftarrow \hat{h}^*$ ;
     $g(i) \leftarrow 0$ ;
    carga  $\leftarrow \mathcal{L}(g)$ ;
    if carga < mín then
      mín  $\leftarrow$  carga;
      collapse  $\leftarrow i$ ;
  if collapse is not None then
     $\hat{h}^*(\text{collapse}) \leftarrow 0$ ;
    remain  $\leftarrow$  remain  $- \{i\}$ ;
  else
    terminar  $\leftarrow \text{True}$ 

```



(a) $\mathcal{D} = 4, \mathcal{L} = 730$



(b) $\mathcal{D} = 2, \mathcal{L} = 100, m_\ell = 2$.

Figura 5: (a) descomposición densa vs. (b) dispersa subóptima para un grafo de 20 nodos con $\mathcal{D}_{\text{ref}} = 5, \mathcal{L}_{\text{ref}} = 100$. En color y tamaño: los alcances; en línea punteada: los enlaces.

mayor alcance y disminuyendo notablemente la superposición de los mismos. Esto se evidencia en los valores de las métricas que se reducen considerablemente, y manteniendo reducido el número de enlaces vinculantes.

Por otro lado, en Fig. 7 y Fig. 8 se observan los valores típicos de los parámetros de interés para la descomposición densa y dispersa, respectivamente. Los resultados fueron generados para un conjunto de 10 frameworks por cada valor de $3 \leq n \leq 100$. Podemos observar cómo en el caso del retardo máximo, el valor de \mathcal{D} se vuelve prácticamente constante e igual a 2 en el caso disperso, indicando que en promedio, no hay subframeworks de tamaño mayor a 1. En el caso de la carga de comunicaciones, vemos que se logra que se mantenga prácticamente igual al valor de referencia, manteniendo reducido el porcentaje de enlaces vinculantes.

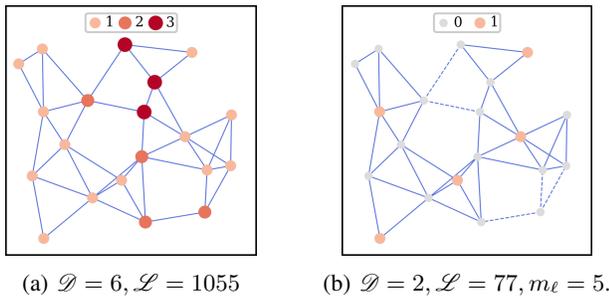


Figura 6: (a) descomposición densa vs. (b) dispersa subóptima para un grafo de 20 nodos con $\mathcal{D}_{\text{ref}} = 5$, $\mathcal{L}_{\text{ref}} = 80$.

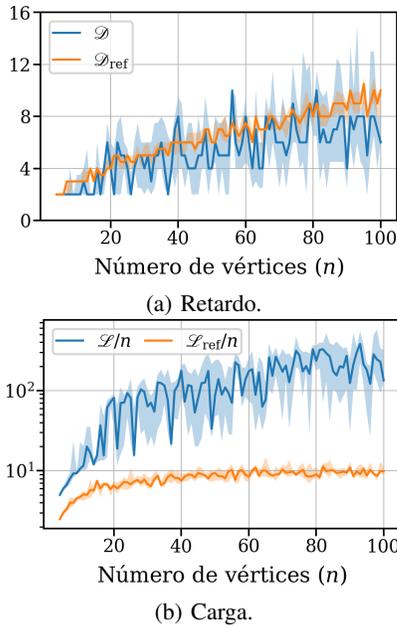


Figura 7: Comparación de las métricas típicas de la descomposición densa vs. los valores de referencia.

VI. CONCLUSIÓN

Este trabajo permitió comparar cualitativa y cuantitativamente distintos tipos de descomposición, aportando a una mejor comprensión este problema. A partir del análisis presentado, se establecieron los criterios de optimalidad y los algoritmos que permitieron seleccionar descomposiciones óptimas aproximadas. Mediante simulaciones computacionales, se pudo observar las claras ventajas de la descomposición dispersa respecto de la densa, situando a la primera como la elegida para futuras investigaciones.

REFERENCIAS

- [1] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, 2013.
- [2] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *Journal of Intelligent & Robotic Systems*, vol. 102, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:233256952>

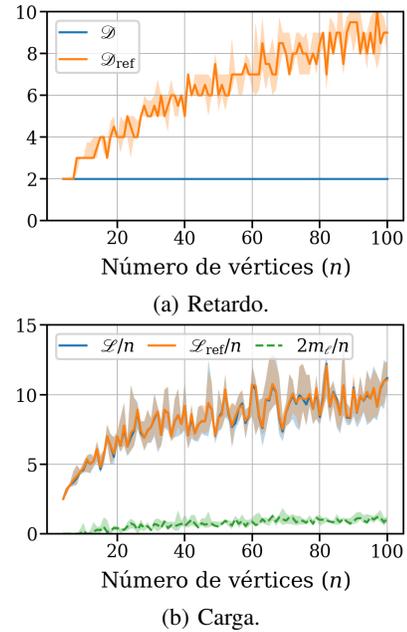


Figura 8: Comparación de las métricas típicas de la descomposición dispersa vs. los valores de referencia. La línea sólida: mediana, área coloreada cuartiles Q_1 (25%) y Q_3 (75%).

- [3] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.
- [4] B. D. O. Anderson, B. Fidan, C. Yu, and D. Walle, "Uav formation control: Theory and application," in *Recent Advances in Learning and Control*, V. D. Blondel, S. P. Boyd, and H. Kimura, Eds. London: Springer London, 2008, pp. 15–33.
- [5] D. Zelazo, A. Franchi, P. Allgöwer, H. Bühlhoff, and P. Giordano, "Rigidity maintenance control for multi-robot systems," in *2012 Robotics: Science and Systems*, 2012.
- [6] D. Zelazo, A. Franchi, H. H. Bühlhoff, and P. R. Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.
- [7] Z. Sun, C. Yu, and B. D. O. Anderson, "Distributed optimization on proximity network rigidity via robotic movements," in *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 6954–6960.
- [8] R. K. Williams, A. Gasparri, A. Priolo, and G. S. Sukhatme, "Distributed combinatorial rigidity control in multi-agent networks," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 6061–6066.
- [9] A. M. Amani, G. Chen, M. Jalili, X. Yu, and L. Stone, "Distributed rigidity recovery in distance-based formations using configuration lattice," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1547–1558, 2020.
- [10] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen, "Graph diameter, eigenvalues, and minimum-time consensus," *Automatica*, vol. 50, no. 2, pp. 635–640, 2014.
- [11] J. F. Presenza, J. I. Alvarez-Hamelin, I. Mas, and J. I. Giribet, "Subframework-based rigidity control in multirobot networks," in *American Control Conference, Atlanta, Georgia, USA, 2022*.
- [12] J. F. Presenza, J. I. Giribet, J. I. Alvarez-Hamelin, and I. Mas, "Control de rigidez por subframeworks en sistemas multirobot," in *Jornadas Argentinas de Robótica, Bariloche, Argentina, 2022*, 2022.
- [13] L. Asimov and B. Roth, "The rigidity of graphs, ii," *Journal of Mathematical Analysis and Applications*, vol. 68, pp. 171–190, 1979.
- [14] J. F. Presenza, "Redes reconfigurables de vehículos no tripulados," Ph.D. dissertation, Universidad de Buenos Aires, 2023.