PESQUIMAT 24(1): 57–69 (2021) ISSN:1560-912X/ ISSN-E:1609-8439
https://doi.org/10.15381/pesquimat.v24i1.20428 Facultad de Ciencias Matemáticas – UNMSM

Introducción a la teoría de complejidad topológica

Cesar A. Ipanaque Zapata [1] *y Rodolfo José Gálvez Pérez*[2]

Resumen: En este trabajo revisaremos la noción de complejidad topológica, introducida por Michael Farber en el 2003. Usaremos esta teoría de complejidad topológica para resolver el problema de planificación de movimiento de un robot móvil que navega en el plano euclidiano evitando colisionar con un obstáculo. Específicamente, calculamos la complejidad topológica y diseñamos algoritmos óptimos.

Palabras clave: Complejidad topológica, Problema de planificación de movimiento, Algoritmos.

Introduction to the topological complexity theory

Abstract: In this work we will review the notion of topological complexity, introduced by Michael Farber in 2003. We will use this theory of topological complexity to solve the motion planning problem of a mobile robot that navigates in the Euclidean plane avoiding colliding with an obstacle. Specifically, we calculate topological complexity and design optimal algorithms.

Keywords: Topological complexity, Motion planning problem, Algoritms.

Recibido: 18/05/2021. *Aceptado:* 10/06/2021. *Publicado online:* 30/06/2021.

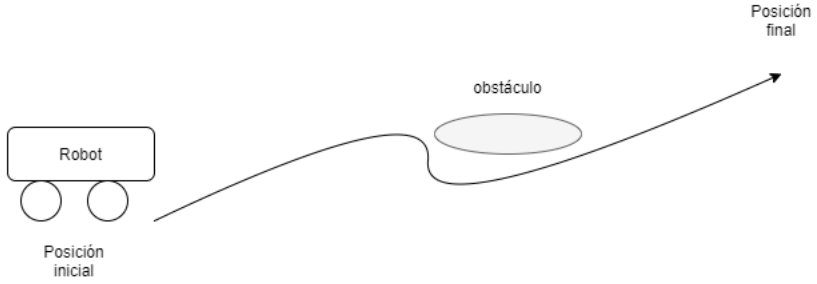
[1]USP, Instituto de Ciências Matemáticas e de Computação, ICMC-USP. e-mail: cesarzapata@usp.br The first author would like to thank grant#2016/18714-8, São Paulo Research Foundation (FAPESP) for financial support.

[2]UNMSM, Facultad de Ciencias Matemáticas, e-mail:rgalvezp@unmsm.edu.pe

1. Introducción

El *problema de planificación de movimiento* (vamos a decir simplemente MPP por sus siglas en ingles de *Motion planning problem*) de robots consiste en encontrar o diseñar algoritmos que planifiquen por completo el movimiento de un sistema mecánico dado, es decir, dada una posición inicial y una posición final el algoritmo da una ruta para que el robot pueda seguir o navegar de forma autónoma desde su posición inicial hasta la posición deseada (final). Este es un problema clásico en el campo de la robótica, el lector puede ver las siguientes referencias [12], [13], [2]. Como ejemplo práctico para aplicar la teoría de complejidad topológica, vamos a considerar un sistema mecánico conformado por un robot móvil, que navega en el plano, y un obstáculo, como muestra la Figura 1.

Figura 1: Sistema mecánico conformado por un robot móvil, que navega en el plano, y un obstáculo. Además, se muestra una ruta para que el robot navegue desde una posición inicial para una posición final sin chocar con el obstáculo.



Michael Farber en el 2003, en [8], presenta un enfoque topológico para el MPP. Sea X el espacio de estados o configuraciones (libre de obstáculos) de un cierto sistema mecánico, el cual es un espacio topológico. Denotemos por $\mathrm{P}X$ el espacio formado por todos los caminos continuos $\gamma : [0,1] \to X$ en X. $\mathrm{P}X$ tiene la topología compacto-abierta. Note que un camino en el espacio de configuraciones X da un movimiento continuo de nuestro robot sin tocar los obstáculos, o sea, una navegación segura de nuestro robot.

Una forma de planificar el movimiento de un sistema mecánico, y así solucionar el MPP, es encontrar un algoritmo s que dé caminos en su espacio de configuraciones X, específicamente, dada una configuración inicial $A \in X$ y una configuración final $B \in X$, s da un camino $s(A,B) : [0,1] \to X$ tal que $s(A,B)(0) = A$ y $s(A,B)(1) = B$. Así s da una ruta para que nuestro robot pueda navegar sin chocar con los obstáculos desde una posición inicial hasta una posición deseada (final).

Como X es un espacio topológico entonces se puede hablar de cercanía. Si $A' \in X$ está cerca de A y $B' \in X$ está cerca de B se debe esperar que $s(A',B')$ este cerca de $s(A,B)$. En términos matemáticos se dice que s dependa continuamente de las variables A e B. En términos de robótica se dice que s tenga estabilidad.

Farber notó que un algoritmo de planificación de movimiento sobre X es una sección de la aplicación

$$e : \mathrm{P}X \to X \times X,\ e(\gamma) = (\gamma(0), \gamma(1)).$$

En otras palabras, es una aplicación $s : X \times X \to PX$ no necesariamente continua tal que

$$s(A,B)(0) = A \text{ y } s(A,B)(1) = B \text{ para todo } (A,B) \in X \times X.$$

Para efectos de estabilidad lo conveniente es que los algoritmos sean continuos. Sin embargo, se puede mostrar que existe un algoritmo continuo sobre X si, y solamente si, X es contráctil.

Por lo tanto, si X no es contráctil entonces cualquier algoritmo sobre X no es continuo. De esta manera, Farber define un invariante numérico, llamado complejidad topológica $\mathrm{TC}(X)$, que

mide las discontinuidades de los algoritmos en X, o también, se dice que mide la complejidad del MPP en X.

En este trabajo vamos usar la teoría de complejidad topológica para solucionar el problema de planificación de movimiento del sistema mecánico dado en la Figura 1, es decir, vamos a calcular su complejidad topológica y diseñar algoritmos óptimos. De esta manera, este trabajo espera contribuir en dar una introducción a la teoría de complejidad topológica para una mayor cantidad de lectores.

2. Complejidad topológica

En esta sección daremos una revisión de la teoría de complejidad topológica. Específicamente responderemos las siguientes preguntas: ¿Qué es? ¿Cómo se usa? ¿Cuales son los problemas centrales? ¿Cómo se calcula?

2.1. ¿Qué es?

La noción de espacio de estados o configuraciones libre de obstáculos aparece en el campo de la Robótica, vea [14]. Para unificar este concepto, vamos a decir que el *espacio de estados* o *configuraciones libre de obstáculos*[1] de un sistema mecánico conformado por un objeto (ejemplo un robot) dentro de un ambiente con obstáculos o no, es el conjunto cuyos elementos determinan la posición de cada punto del objeto sin colisionar con los obstáculos. Generalmente, el espacio de estados es un espacio topológico (por ejemplo en Robótica, los espacios de estados son subespacios de algún espacio Euclidiano $\mathbb{R}^d$, donde cada coordenada de una d-tupla representan los parámetros que caracterizan la posición de cada punto en el robot sin colisionar con los obstáculos). Así, por definición del espacio de estados, un camino en el espacio de estados o configuraciones da un movimiento continuo de nuestro objeto sin tocar los obstáculos.

Sea X el espacio de estados o configuraciones de un cierto sistema mecánico. Denotemos por $\mathrm{P}X$ el espacio formado por todos los caminos continuos $\gamma : [0,1] \to X$ en X. $\mathrm{P}X$ es dotado de la topología compacto-abierta[2]. Recuerde que un camino en el espacio de configuraciones X da un movimiento continuo de nuestro robot sin tocar los obstáculos, o sea, una navegación segura de nuestro robot.

Una forma de planificar el movimiento de un sistema mecánico, y así solucionar el MPP, es encontrar un *algoritmo*, digamos s, que dé caminos en su espacio de configuraciones X, específicamente, dada una configuración inicial $A \in X$ y una configuración final $B \in X$, s da un camino $s(A,B) : [0,1] \to X$ tal que $s(A,B)(0) = A$ y $s(A,B)(1) = B$. Así s da una ruta para que nuestro robot pueda navegar sin chocar con los obstáculos desde una posición inicial hasta una posición deseada (final).

Denotemos por $1_X : X \to X$ a la aplicación identidad. Note que un algoritmo en X es una *sección* o *inversa a derecha* de la aplicación

$$e : \mathrm{P}X \to X \times X,\ e(\gamma) = (\gamma(0), \gamma(1)). \tag{2.1}$$

En otras palabras, es una aplicación $s : X \times X \to PX$ no necesariamente continua tal que $e \circ s = 1_{X\times X}$, osea,

$$s(A,B)(0) = A \text{ y } s(A,B)(1) = B \text{ para todo } (A,B) \in X \times X. \tag{2.2}$$

En este caso diremos que s es un *algoritmo de planificación de movimiento* en X. Un algoritmo s es continuo si la aplicación $s : X \times X \to \mathrm{P}X$ es continua.

[1] Simplemente diremos espacio de estados o configuraciones.

[2] Para una revisión de la topología compacto-abierta el lector puede consultar [1].

Recordemos que un espacio topológico X es llamado *contráctil* si existe una aplicación continua $H : X \times [0,1] \to X$ tal que $H_0 = 1_X$ e $H_1 = \overline{x_0}$ para alguna constante $x_0 \in X$. Donde H_t denota la aplicación $H_t : X \to X$ dada por $H_t(x) = H(x,t)$, para cualquier $x \in X$, $1_X : X \to X$ es la aplicación identidad y $\overline{x_0}$ denota la aplicación constante en x_0.

Para efectos de estabilidad en los algoritmos es conveniente que los algoritmos sean continuos. Sin embargo, Farber muestra que existe un algoritmo continuo en X si X es contráctil. La demostración del Lema 2.1 es técnica y será usada para construir algoritmos sobre espacios contráctiles, así que presentamos una demostración conveniente para nuestros propósitos.

Lema 2.1. [8, Theorem 1, pg. 212] Sea X un espacio topológico. Tenemos que existe un algoritmo continuo en X si y solamente si X es contráctil.

Demostración. ($\Rightarrow$) Sea $s : X \times X \to \mathrm{P}X$ un algoritmo continuo en X, o sea, s es una aplicación continua que satisface las siguientes igualdades: $s(x_1,x_2)(0) = x_1$ y $s(x_1,x_2)(1) = x_2$, para cualquier $(x_1,x_2) \in X \times X$. Veamos que X es contráctil. De hecho, fijemos un $x_0 \in X$ y definamos la aplicación $H : X \times [0,1] \to X$ por

$$H(x,t) = s(x,x_0)(t), \text{ para cualquier } (x,t) \in X \times [0,1].$$

Como s es continua, sigue que H es continua. Además, para $x \in X$ obtenemos que

$$\begin{aligned} H(x,0) &= s(x,x_0)(0) \\ &= x. \end{aligned}$$

Similarmente, para $x \in X$ obtenemos que $H(x,1) = x_0$. Por lo tanto, X es contráctil.

($\Leftarrow$) Sea $H : X \times [0,1] \to X$ una aplicación continua tal que $H_0 = 1_X$ e $H_1 = \overline{x_0}$, para alguna constante $x_0 \in X$. Veamos que existe un algoritmo continuo en X. De hecho, definamos la aplicación $s : X \times X \to \mathrm{P}X$ dada por

$$s(x_1,x_2)(t) = \begin{cases} H(x_1,2t), & \text{si } 0 \leq t \leq 1/2, \\ H(x_2,2-2t), & \text{si } 1/2 \leq t \leq 1, \end{cases} \text{ para cualquier } (x_1,x_2) \in X \times X.$$

Como H es continua, s es continua. Además, para $(x_1,x_2) \in X \times X$, tenemos que

$$\begin{aligned} s(x_1,x_2)(0) &= H(x_1,0) \\ &= x_1, \end{aligned}$$

similarmente, tenemos que $s(x_1,x_2)(1) = H(x_2,0) = x_2$. Por lo tanto, s es un algoritmo continuo en X. □

Usaremos la demostración del Lema 2.1 para construir un algoritmo continuo en espacios estrellados. Recordemos que un subconjunto $K \subset \mathbb{R}^d$ es llamado *estrellado* si existe un punto $x_0 \in K$ tal que los puntos $(1-t)x + tx_0 \in K$, para cualquier $x \in K$ y cualquier $t \in [0,1]$. En ese caso, x_0 es llamado la *estrella* de K.

Ejemplo 2.2 (Algoritmo continuo en espacios estrellados)**.** Sea K un conjunto estrellado cuya estrella es $x_0 \in K$. La aplicación $H : K \times [0,1] \to K$ dada por

$$H(x,t) = (1-t)x + tx_0$$

es una homotopia satisfaciendo $H_0 = 1_K$ y $H_1 = \overline{x_0}$. Por lo tanto, K es contráctil. Además, por la demostración del Lema 2.1 obtenemos que la aplicación $s : K \times K \to \mathrm{P}K$ dada por

$$s(x_1,x_2)(t) = \begin{cases} (1-2t)x_1 + 2tx_0, & \text{si } 0 \leq t \leq 1/2, \\ (2t-1)x_2 + (2-2t)x_0, & \text{si } 1/2 \leq t \leq 1, \end{cases} \text{ para cualquier } (x_1,x_2) \in K \times K,$$

es un algoritmo continuo en K.

El Lema 2.1 implica que si X no es contráctil entonces cualquier algoritmo en X no es continuo. De esta manera, Farber define un invariante numérico, llamado complejidad topológica $\mathrm{TC}(X)$, que mide las discontinuidades de los algoritmos en X, o también se dice que mide la complejidad del MPP de un sistema mecánico cuyo espacio de estados es X.

Definición 2.3. [8, Definition 2, pg. 213] La *complejidad topológica* de un espacio topológico X, denotado por $\mathrm{TC}(X)$, es el menor entero positivo m tal que el producto cartesiano $X \times X$ puede ser cubierto por m subconjuntos abiertos U_i,

$$X \times X = U_1 \cup U_2 \cup \cdots \cup U_m,$$

tal que para cada $i = 1, 2, \ldots, m$, existe una aplicación continua $s_i : U_i \to \mathrm{P}X$ tal que $s_i(x_1, x_2)(0) = x_1$ y $s_i(x_1, x_2)(1) = x_2$ para cualquier $(x_1, x_2) \in U_i$. En ese caso, s_i es llamado un *algoritmo local* continuo sobre U_i. Se tal m no existe diremos que $\mathrm{TC}(X) = \infty$.

Note que $\mathrm{TC}(X) = 1$ si y solamente si existe un algoritmo continuo en X. Usando el Lema 2.1, obtenemos que es posible planificar, mediante un algoritmo estable, el movimiento de un sistema mecánico si y solamente si su espacio de estados X es contráctil. Así, $\mathrm{TC}(X) = 1$ si y solamente si X es contráctil. Esto explica, el porqué en las aplicaciones industriales no es posible encontrar un algoritmo estable que planifique el movimiento, ya que los espacios de estados de sistemas mecánicos que mayormente aparecen en la industria no son contráctiles.

Notemos que, toda colección $s = \{s_i : U_i \to \mathrm{P}X\}_{i=1}^k$ con cada $U_i \subset X \times X$ abierto, $X \times X = \bigcup_{i=1}^k U_i$ y $e \circ s_i = incl_{U_i}$, donde $incl_{U_i} : U_i \hookrightarrow X \times X$ denota la aplicación inclusión, define un algoritmo $s : X \times X \to \mathrm{P}X$ en X. De hecho, para cada $(x_1, x_2) \in X \times X$ elegimos el menor $i \in \{1, \ldots, k\}$ tal que $(x_1, x_2) \in U_i$ y definimos el camino $s(x_1, x_2)$ por $s_i(x_1, x_2)$. Un tal algoritmo $s = \{s_i : U_i \to PX\}_{i=1}^k$ es llamado *óptimo* se $k = \mathrm{TC}(X)$.

Por tanto, la complejidad topológica $\mathrm{TC}(X)$ es un invariante numérico que mide la complejidad del MPP para un sistema mecánico cuyo espacio de estados es X. En otras palabras, la complejidad topológica $\mathrm{TC}(X)$ da la menor cantidad de algoritmos locales continuos que se necesitan para planificar el movimiento de un sistema mecánico cuyo espacio de estados es X.

2.2. ¿Cómo se usa?

Consideremos un sistema mecánico cuyo espacio de estados es X. Si su complejidad topológica $\mathrm{TC}(X) = k$ sabemos que existe un algoritmo óptimo $s = \{s_i : U_i \to \mathrm{P}X\}_{i=1}^k$ en X. Así, si damos un estado inicial x_1 y un estado final x_2, el algoritmo da una ruta o un movimiento continuo $s(x_1, x_2)$, empezando en x_1 y terminando en x_2, para que el robot navegue de forma segura y autónoma. De esta manera, con la teoría de complejidad topológica, no únicamente planificamos el movimiento, sino también planificamos el movimiento de manera óptima, o sea, con la menor cantidad de algoritmos locales. En particular, el invariante $\mathrm{TC}(X)$ da la menor cantidad de algoritmos locales que debe tener cualquier algoritmo en X.

2.3. ¿Cuales son los problemas centrales?

Calcular la complejidad topológica TC es un problema difícil y propio de la topología algebraica. Mayormente, se desarrollan cotas inferiores y superiores que permitan estimar el valor de la TC (vea la Sección 2.4). Sin embargo, esas técnicas no permiten construir o diseñar los algoritmos. Así que, podemos decir que los problemas centrales de la teoría de complejidad topológica son los siguientes:

- Calcular $\mathrm{TC}(-)$.
- Encontrar algoritmos óptimos.

El problema de calcular la TC es muy estudiado en el área de la topología algebraica. Las herramientas principales usadas para estimar la TC son: Categoría de Lusternik-Schnirelmann, teorías de cohomología, teoría de homotopía, complejos CW, fibraciones y cofibraciones, teoría de obstrucción, teoría de homotopía racional, etc. El lector puede consultar las siguientes referencias [8], [15], [4], [5], [6], [11] y las referencias de ellos.

El problema de encontrar o diseñar algoritmos óptimos es propio de la topología y geometría. Este es un problema poco estudiado y en general independiente del problema de calcular la TC. El lector puede consultar los trabajos [17], [16] y las referencias de ellos.

Observación 2.4. Un problema extra y útil para fines prácticos seria la implementación, en algún programa computacional, de los algoritmos obtenidos vía la teoría de complejidad topológica. Sin embargo, este problema aún no ha sido estudiado.

2.4. ¿Cómo se calcula?

Calcular la TC es un problema difícil y propio de la topología algebraica. Una propiedad muy útil es que la TC es un invariante homotópico. Presentamos la demostración de este hecho, ya que será usada en la construcción de algoritmos.

Recordemos que dos espacios topológicos X y Y *tienen el mismo tipo de homotopía* si existen aplicaciones continuas $f : X \to Y$ y $g : Y \to X$ tales que $f \circ g \simeq 1_Y$ y $g \circ f \simeq 1_X$.

Proposición 2.5. Si existen aplicaciones continuas $Y \xrightarrow{g} X \xrightarrow{f} Y$ tales que $f \circ g \simeq 1_Y$, entonces vale la siguiente desigualdad:

$$\mathrm{TC}(X) \geq \mathrm{TC}(Y).$$

En particular, si X y Y tienen el mismo tipo de homotopía, entonces $\mathrm{TC}(X) = \mathrm{TC}(Y)$.

Demostración. Sea $H : Y \times [0,1] \to Y$ una homotopía tal que $H_0 = 1_Y$ y $H_1 = f \circ g$. Note que, todo algoritmo local $s : U \to \mathrm{P}X$ definido sobre $U \subset X \times X$ induce un algoritmo local $\widetilde{s} : V \to \mathrm{P}Y$, definido sobre $V = (g \times g)^{-1}(U) \subset Y \times Y$, dado por:

$$\widetilde{s}(y_1, y_2)(t) = \begin{cases} H_{3t}(y_1), & \text{si } 0 \leq t \leq 1/3; \\ f\left(s(g(y_1), g(y_2))(3t-1)\right), & \text{si } 1/3 \leq t \leq 2/3; \\ H_{3-3t}(y_2), & \text{si } 2/3 \leq t \leq 1. \end{cases} \tag{2.3}$$

Por lo tanto, si $s = \{s_i : U_i \to \mathrm{P}X\}_{i=1}^k$ es un algoritmo óptimo en X, entonces aplicando (2.3) a cada s_i, podemos construir un algoritmo (no necesariamente óptimo) $\widetilde{s} = \{\widetilde{s_i} : V_i \to \mathrm{P}Y\}_{i=1}^k$ en Y. Así, obtenemos que $\mathrm{TC}(X) = k \geq \mathrm{TC}(Y)$.

Además, si X y Y tienen el mismo tipo de homotopía, podemos obtener de forma similar la otra desigualdad y así $\mathrm{TC}(X) = \mathrm{TC}(Y)$. Note que, en este caso el algoritmo $\widetilde{s}$ es óptimo. □

Para calcular la TC, en general se intenta estimar, o sea, encontrar cotas inferiores y superiores. Las cotas inferiores mayormente dependen de la información algebraica del espacio, por ejemplo, de su anillo de cohomología singular $H^*(X)$ ya sea con coeficientes enteros $\mathbb{Z}$ o sobre un cuerpo $\mathbb{K}$. Las cotas superiores aparecen de la información homotópica del espacio, por ejemplo, de su dimensión homotópica y de su grado de conexidad. Continuación enunciaremos las cotas más usadas para estimar la TC.

Recordemos primero, de [7], que la *categoría de Lusternik-Schnirelmann* de un espacio topológico X, denotado por $\mathrm{cat}(X)$, es el menor entero positivo k tal que X puede ser cubierto por subconjuntos abiertos $U_1, \ldots, U_k$, o sea, $X = \bigcup_{i=1}^k U_i$, tales que cada inclusión $U_i \hookrightarrow X$ sea homotópica a una aplicación constante. Note que, $\mathrm{cat}(X) = 1$ si y solamente si X es contráctil.

Para un anillo A y $S \subset A$ un subconjunto, el *índice de nilpotencia* de S en A es definido por:

$$\mathrm{Nil}(S) = \mathrm{mín}\{k : \quad \text{cualquier producto de } k \text{ elementos de } S \text{ es nulo}\}.$$

Para un complejo CW X, su *dimensión homotópica* es dado por

$$\mathrm{hdim}(X) = \mín\{\dim(Y) :\ Y \text{ es un complejo CW y } Y \simeq X\}.$$

Un espacio topológico X es q-*conexo* si $\pi_i(X) = 0$ para cualquier $0 \leq i \leq q$.

La demostración del siguiente teorema es propio de la topología algebraica y está fuera del objetivo de este trabajo. El lector interesado puede consultar las referencias indicadas en cada afirmación.

Teorema 2.6. 1) [8, Theorem 5, pg. 215] Para cualquier espacio conexo por caminos X, se tiene:

$$\mathrm{cat}(X) \leq \mathrm{TC}(X) \leq \mathrm{cat}(X \times X).$$

2) [7, Theorem 1.37, pg. 18], [8, Theorem 11, pg. 218] Si X y Y son complejos CW, entonces

$$\begin{aligned} \mathrm{cat}(X \times Y) &\leq \mathrm{cat}(X) + \mathrm{cat}(Y) - 1, \\ \mathrm{TC}(X \times Y) &\leq \mathrm{TC}(X) + \mathrm{TC}(Y) - 1. \end{aligned}$$

3) [7, Theorem 1.50, pg. 23] Si X es un complejo CW y q-conexo, entonces

$$\begin{aligned} \mathrm{cat}(X) &\leq \frac{\mathrm{hdim}(X)}{q+1} + 1, \\ \mathrm{TC}(X) &\leq \frac{2\mathrm{hdim}(X)}{q+1} + 1. \end{aligned} \tag{2.4}$$

4) [8, Theorem 7, pg. 216] Si h^* es cualquier teoría de cohomología multiplicativa sobre los pares de espacios topológicos, entonces

$$\mathrm{Nil}\left(\mathrm{Ker}(\Delta^* : h^*(X \times X) \to h^*(X))\right) \leq \mathrm{TC}(X). \tag{2.5}$$

Donde $\Delta : X \to X \times X,\ \Delta(x) = (x, x)$ es la aplicación diagonal.

Ejemplo 2.7 (Complejidad topológica de las esferas impares)**.** Consideremos la esfera $(2m-1)$-dimensional $S^{2m-1} = \{x \in \mathbb{R}^{2m} :\ \| x \| = 1\}$, con $m \geq 1$, donde $\| \cdot \|$ denota la norma euclidiana. Veamos que su complejidad topológica es igual a 2, o sea,

$$\mathrm{TC}(S^{2m-1}) = 2, \text{ para cualquier } m \geq 1.$$

De hecho, como S^{2m-1} no es contráctil, entonces tenemos que $\mathrm{TC}(S^{2m-1}) \geq 2$. Así, basta mostrar que $\mathrm{TC}(S^{2m-1}) \leq 2$. Para ello, vamos construir un algoritmo $s = \{s_i : U_i \to \mathrm{P}S^{2m-1}\}_{i=1}^{2}$ en S^{2m-1} de la siguiente manera:

- Consideremos un campo vectorial tangente sobre la esfera S^{2m-1}, por ejemplo, el campo $\nu : S^{2m-1} \to S^{2m-1}$ dado por $\nu(x_1, y_1, \ldots, x_m, y_m) = (-y_1, x_1, \ldots, -y_m, x_m)$.

- Los abiertos U_i son dados por:

$$\begin{aligned} U_1 &= \{(a, b) \in S^{2m-1} \times S^{2m-1} :\ a \neq -b\}, \\ U_2 &= \{(a, b) \in S^{2m-1} \times S^{2m-1} :\ a \neq b\}. \end{aligned}$$

  Note que, $U_1 \cup U_2 = S^{2m-1} \times S^{2m-1}$.

- El algoritmo local $s_1 : U_1 \to \mathrm{P}S^{2m-1}$ es dado por:

$$s_1(a, b)(t) = \frac{(1-t)a + tb}{\| (1-t)a + tb \|}, \text{ para todo } (a, b) \in U_1 \text{ y } t \in [0, 1].$$

  Note que, fijados $(a, b) \in U_1$, o sea, a y b no son antípodas, el camino $s_1(a, b)(-)$ es la geodésica más corta en la esfera S^{2m-1} que conecta a y b.

- Antes de definir el segundo algoritmo local s_2 consideremos el subconjunto $F = \{(a,b) \in S^{2m-1} \times S^{2m-1} : \quad a = -b\}$ y para cualquier $(a,b) \in F$ definamos

$$\alpha(a,b)(t) = \begin{cases} s_1(a, v(a))(2t), & \text{si } 0 \leq t \leq 1/2; \\ s_1(v(a), b)(2t-1), & \text{si } 1/2 \leq t \leq 1. \end{cases}$$

  Note que, $\nu(a)$ no es antípoda de a ni de b, pues $\nu(a)$ es ortogonal a a y a $b = -a$.

- Ahora, definamos $s_2 : U_2 \to \mathrm{P}S^{2m-1}$. Para cualquier, $(a,b) \in U_2$, sea

$$s_2(a,b)(t) = \begin{cases} s_1(a, -b)(2t), & \text{si } 0 \leq t \leq 1/2; \\ \alpha(-b, b)(2t-1), & \text{si } 1/2 \leq t \leq 1. \end{cases}$$

  Note que $-b$ no es antípoda de a, pues $a \neq b$.

Así, tenemos que $\mathrm{TC}(S^{2m-1}) = 2$. Además, tal algoritmo $s = \{s_i : U_i \to \mathrm{P}S^{2m-1}\}_{i=1}^{2}$ es óptimo.

□

Observación 2.8. Note que, usando la desigualdad (2.4) del Teorema 2.6 obtenemos la siguiente cota superior: $\mathrm{TC}(S^d) \leq 3$, para cualquier $d \geq 1$, pues $\mathrm{hdim}(S^d) = d$ y la esfera S^d es $(d-1)$-conexa. Queremos resaltar que para calcular la TC de la esfera S^{2m-1} hemos construido explícitamente un algoritmo y así hemos obtenido una mejor cota superior, $\mathrm{TC}(S^{2m-1}) \leq 2$.

Por otro lado, usando el anillo de cohomología $\mathbb{Q}$-singular $H^*(S^d; \mathbb{Q}) = \dfrac{\mathbb{Q}[\alpha]}{\langle \alpha^2 \rangle}$ en la desigualdad (2.5) del Teorema 2.6 se puede obtener que $3 \leq \mathrm{TC}(S^d)$, para d par (El lector interesado puede consultar [8, pag. 216]). Así, $\mathrm{TC}(S^d) = 3$, para cualquier d par.

3. Aplicación

En esta sección vamos usar la teoría de complejidad topológica para solucionar el problema de planificación de movimiento para el sistema mecánico dado en la Figura 1, es decir, vamos calcular la complejidad topológica y diseñar algoritmos óptimos.

Consideremos que nuestro robot móvil tiene un radio $l_R > 0$ y que el obstáculo tiene un radio $l_O > 0$. Además, consideremos que nuestro sistema de referencia $\mathbb{R}^2$ tiene su origen de coordenadas en el centro del obstáculo (vea la Figura 2). Para encontrar el espacio de estados (libre de obstáculos) de nuestro sistema mecánico, escojamos un punto de nuestro robot, por ejemplo, su centro de gravedad proyectado verticalmente sobre el plano, y denotemos por p_R (como muestra la Figura 2). Note que, si $p_R \in \mathbb{R}^2$ es tal que $\| p_R \| > l_O + l_R$ entonces para tal p_R nuestro robot está en una posición sin tocar el obstáculo (vea la Figura 2).

Así, tal p_R determina un estado (libre de obstáculos) de nuestro robot, o sea, el espacio de estados X asociado a nuestro sistema mecánico (vea la Figura 3) está dado por:

$$X = \{p_R \in \mathbb{R}^2 : \quad \| p_R \| > l_O + l_R\}.$$

Sin perdida de generalidad, podemos suponer que $l_O + l_R < 1$ (como muestra la Figura 3). Luego, podemos mostrar que X *se retrae por deformación sobre* S^1, o sea, existe una homotopía $H : X \times [0,1] \to X$ satisfaciendo $H_0 = 1_X$, $H_1(X) \subset S^1$ y $H_1(z) = z$, para cualquier $z \in S^1$. Considerando $r = H_1 : X \to S^1$ y $i : S^1 \hookrightarrow X$ la aplicación inclusión, tenemos que $r \circ i = 1_{S^1}$ y $i \circ r \simeq 1_X$. En particular, X tiene el mismo tipo de homotopía que la 1-esfera S^1. De hecho, basta definir

$$H(p_R, t) = (1-t)p_R + t\frac{p_R}{\| p_R \|}, \tag{3.1}$$

para cualquier $(p_R, t) \in X \times [0,1]$.

Así, obtenemos la complejidad topológica para X.

Figura 2: Sistema mecánico conformado por un robot móvil de radio $l_R > 0$, que navega en el plano $\mathbb{R}^2$, y un obstáculo de radio $l_O > 0$. Si $p_R \in \mathbb{R}^2$ es tal que $\| p_R \| > l_O + l_R$ entonces para tal p_R nuestro robot está en una posición sin tocar el obstáculo.

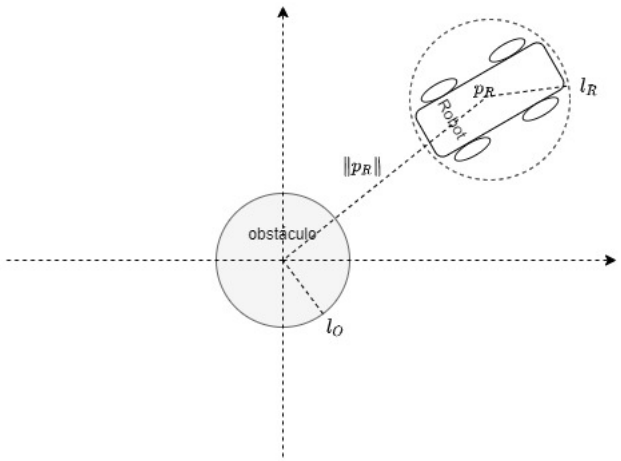
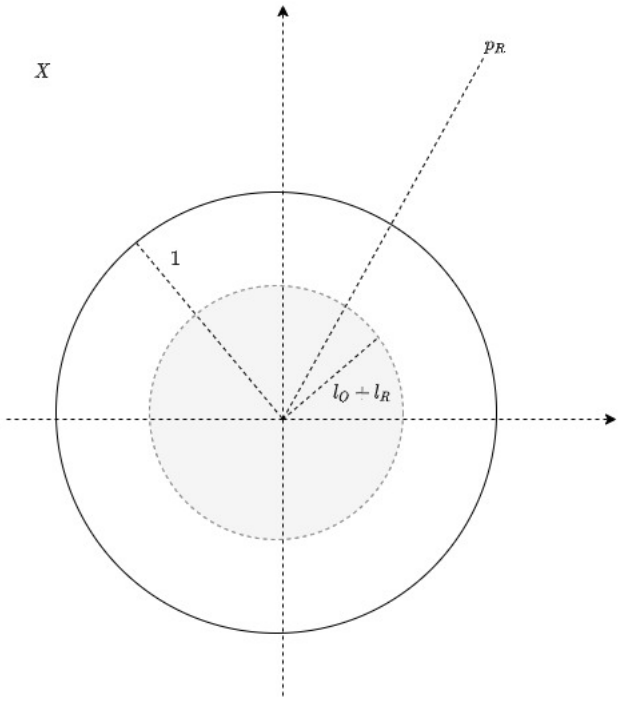


Figura 3: El espacio de estados X.



Proposición 3.1. La complejidad topológica para X es dada por:

$$\mathrm{TC}(X) = 2.$$

Demostración. Como X y S^1 tienen el mismo tipo de homotopía, por la Proposición 2.5, sigue que $\mathrm{TC}(X) = \mathrm{TC}(S^1)$. Usando el Ejemplo 2.7, concluimos que $\mathrm{TC}(X) = 2$. □

Por otro lado, como la homotopía H dada en (3,1) es explicita, entonces, usando la demostración de la Proposición 2.5, tenemos que el algoritmo óptimo s en la esfera S^1, dado en el Ejemplo 2.7, induce un algoritmo óptimo en X.

Proposición 3.2. Existe un algoritmo óptimo explícito en X.

Demostración. Sea $s = \{s_i : U_i \to \mathrm{P}S^1\}_{i=1}^2$ el algoritmo óptimo en S^1, dado en el Ejemplo 2.7. Por la demostración de la Proposición 2.5, tenemos que s induce un algoritmo óptimo $\widetilde{s} = \{\widetilde{s}_i : V_i \to \mathrm{P}X\}_{i=1}^2$ en X. Donde

$$\begin{aligned} V_1 &= (r \times r)^{-1}(U_1) \\ &= \{(z_1, z_2) \in X \times X : \ (r(z_1), r(z_2)) \in U_1\} \\ &= \{(z_1, z_2) \in X \times X : \ \frac{z_1}{\| z_1 \|} \neq -\frac{z_2}{\| z_2 \|}\}. \end{aligned}$$

Similarmente, tenemos que

$$V_2 = \{(z_1, z_2) \in X \times X : \ \frac{z_1}{\| z_1 \|} \neq \frac{z_2}{\| z_2 \|}\}.$$

Los algoritmos locales $\widetilde{s}_i$ están dados por:

$$\widetilde{s}_i(z_1, z_2)(t) = \begin{cases} H_{3t}(z_1), & \text{si } 0 \leq t \leq 1/3; \\ i\left(s_i(r(z_1), r(z_2))(3t-1)\right), & \text{si } 1/3 \leq t \leq 2/3; \\ H_{3-3t}(z_2), & \text{si } 2/3 \leq t \leq 1. \end{cases}$$

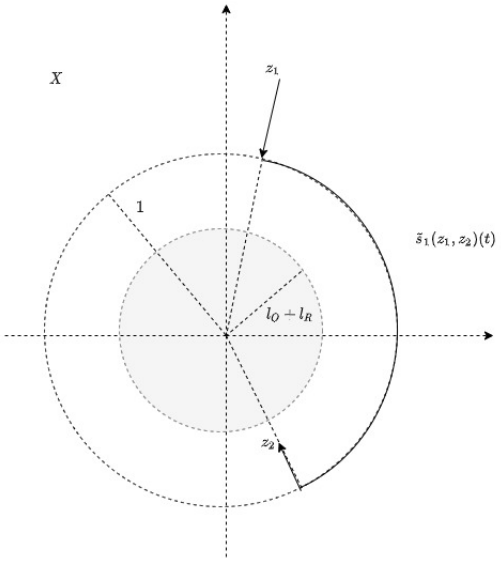


Figura 4: El algoritmo local $\widetilde{s}_1$.

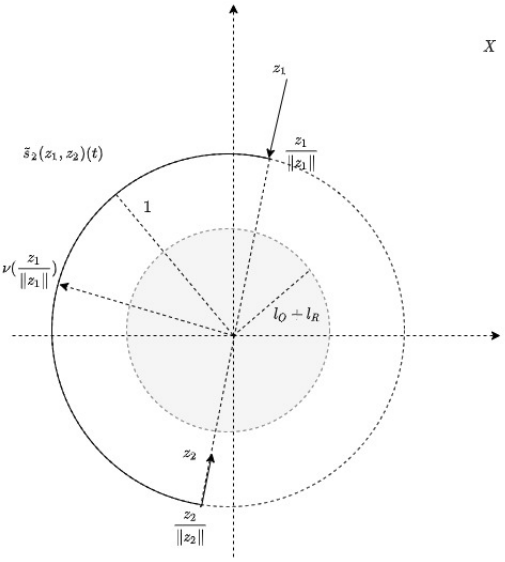


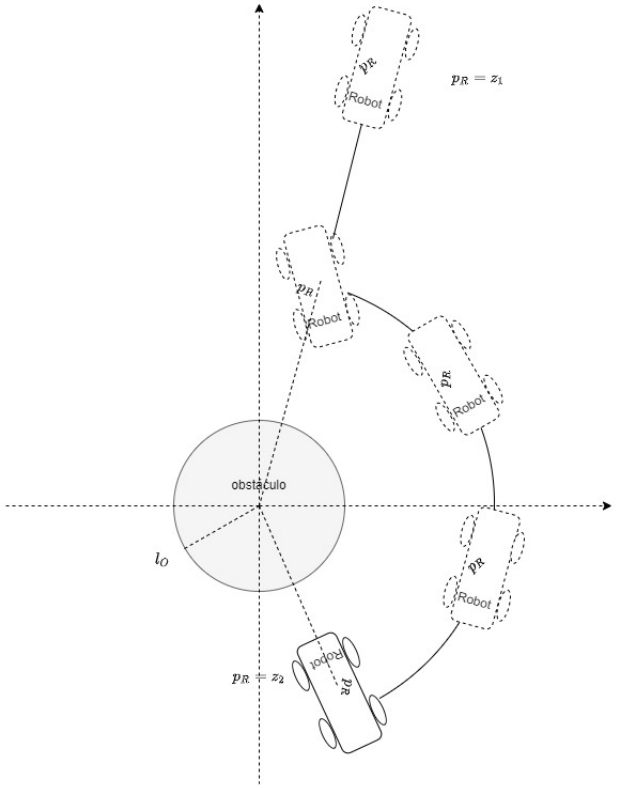
Figura 5: El algoritmo local $\widetilde{s}_2$.

□

Finalmente, en la siguiente Observación, vamos usar el algoritmo $\widetilde{s}$ para planificar el movimiento de nuestro robot, sin colisionar con el obstáculo, desde una posición inicial hasta una posición deseada. Así, $\widetilde{s}$ torna a nuestro robot en un robot autónomo capaz de navegar en el plano sin colisionar con el obstáculo.

Observación 3.3. Note que el camino $\widetilde{s}_1(z_1, z_2)(t)$ dado en la Figura 4 induce un movimiento continuo en nuestro sistema mecánico como muestra la Figura 6.

Figura 6: Movimiento continuo del robot sin colisionar con el obstáculo desde la posición inicial $p_R = z_1$ hacia la posición final $p_R = z_2$ usando la ruta $\widetilde{s}_1(z_1, z_2)(t)$.



4. Conclusión

La teoría de complejidad topológica es un enfoque topológico desarrollado por Farber en el 2003 para solucionar el problema de planificación de movimiento de robots. Hemos dado una introducción básica a esta teoría y para resaltar su importancia hemos resuelto el problema de planificación de movimiento de un sistema mecánico que consiste de un robot móvil que navega en el plano sin colisionar con un obstáculo. Así, este trabajo espera ser una introducción básica a la teoría de complejidad topológica para una mayor cantidad de lectores.

Referencias bibliográficas

[1] Aguilar, M., Gitler, S., and Prieto, C. (2002). *Algebraic Topology from a Homotopical Viewpoint*. UTX, Springer.

[2] Bajd, T., Mihelj, M., Lenarcic, J., Stanovnik, A., and Munih, M. (2010). *Robotics*. International Series on Intelligent Systems, Control, and Automation: Science and Engineering, **43**.

[3] Basabe, I., González, J., Rudyak, Y.B., and Tamaki, D. (2014). *Higher topological complexity and its symmetrization*. Algebr. & Geom. Topol. **14**, no. 4, 2103–2124.

[4] Cohen, D. C. (2018). *Topological complexity of classical configuration spaces and related objects*, in: Topological complexity and related topics, Contemp. Math., vol. 702, pp. 41—60, Amer. Math. Soc., Providence, RI; MR3762831

[5] Cohen, D. C., Farber, M., and Weinberger, S. (2020). *Topology of parametrised motion planning algorithms*. arXiv preprint arXiv:2009.06023.

[6] Cohen, D. C., Farber, M., and Weinberger, S. (2020). *Parametrized topological complexity of collision-free motion planning in the plane*. arXiv preprint arXiv:2010.09809.

[7] Cornea, O., Lupton, G., Oprea, J., and Tanré, D. (2003). *Lusternik-Schnirelmann Category*. Mathematical Surveys and Monographs, 103 (American Mathematical Society, Providence, RI).

[8] Farber, M. (2003). *Topological complexity of motion planning*. Discrete and Computational Geometry. *29* (2), 211–221.

[9] Farber, M. (2017). *Configuration spaces and robot motion planning algorithms*, in: Combinatorial and Toric Homotopy: Introductory Lectures (eds. A. Darby, J. Grbic and J. Wu) (World Scientific, Singapore, 2017), 263–303.

[10] González, J., and Grant, M. (2015). *Sequential motion planning of non-colliding particles in Euclidean spaces*. Proceedings of the American Mathematical Society. **143**, no. 10, 4503–4512.

[11] González, J., Grant, M., and Vandembroucq, L. (2019). *Hopf invariants for sectional category with applications to topological robotics*. The Quarterly Journal of Mathematics, 70(4), 1209-1252.

[12] Latombe, J. C. (1991). *Robot motion planning*. Springer, New York.

[13] LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press, Cambridge.

[14] Lozano-Pérez T. (1990). *Spatial Planning: A Configuration Space Approach*. In: Cox I.J., Wilfong G.T. (eds) Autonomous Robot Vehicles. Springer, New York, NY.

[15] Rudyak, Y. (2010). *On higher analogs of topological complexity.* Topology and its Applications, Elsevier. **157**, no. 5, 916–920.

[16] Zapata, C. A. I., and González, J. (2020). *Multitasking collision-free optimal motion planning algorithms in Euclidean spaces.* Discrete Mathematics, Algorithms and Applications, 12, no. 3. 2050040. doi:10.1142/S1793830920500408

[17] Zapata, C. A. I., and González, J. (2020). *Sequential collision-free optimal motion planning algorithms in punctured Euclidean spaces.* Bulletin of the Australian Mathematical Society, 102(3), 506–516. doi:10.1017/S0004972720000167