

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



**Maestría en Ciencias  
en Ciencias de la Computación**

---

**Algoritmos para exploración de ambientes desconocidos  
utilizando múltiples robots.**

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:

**Luis Enrique Canul Arceo**

Ensenada, Baja California, México

2017

Tesis defendida por

**Luis Enrique Canul Arceo**

y aprobada por el siguiente Comité

---

Dr. Edgar Leonel Chávez González

Codirector del Comité

---

Dr. Ubaldo Ruiz López

Codirector del Comité

Dra. María del Carmen Maya Sánchez

Dr. Carlos Alberto Brizuela Rodríguez



---

Dr. Jesús Favela Vara

Coordinador del Programa de Posgrado en Ciencias de la Computación

---

Dra. Rufina Hernández Martínez

Directora de Estudios de Posgrado

*Luis Enrique Canul Arceo © 2017*

*Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis*

Resumen de la tesis que presenta **Luis Enrique Canul Arceo** como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación

### **Algoritmos para exploración de ambientes desconocidos utilizando múltiples robots.**

Resumen aprobado por:

---

Dr. Edgar Leonel Chávez González

Codirector de Tesis

Dr. Ubaldo Ruiz López

Codirector de Tesis

Desde hace décadas, existe un gran interés por robots que asistan a la humanidad en tareas de la vida diaria. Para lograrlo deben ser capaces de navegar a través de ambientes de forma autónoma. Existen gran variedad de aplicaciones donde se requiere la ayuda de robots para realizar tareas de exploración, inspección y rescate. El resultado de la exploración tiene distintas aplicaciones, entre las que se encuentran la búsqueda de un objeto estático y la verificación de riesgos en ambientes compuesto por múltiples elementos. La exploración de ambientes es un tema que se ha estudiado durante muchos años. Existen gran variedad de trabajos que abordan la exploración de ambientes desconocidos, sin embargo, la mayoría utilizan sensores sofisticados y abordan el problema suponiendo restricciones que muchas veces son difíciles de satisfacer en ambientes reales. La mayor parte de las soluciones propuestas no ofrecen garantías teóricas de que la tarea será completada adecuadamente. El objetivo de este trabajo es garantizar la exploración completa del ambiente bajo ciertas restricciones. En esta tesis suponemos que los robots tienen limitantes de comunicación, que sólo poseen sensores que detectan discontinuidades y no son capaces de obtener su posición dentro del ambiente. Se proponen algoritmos para la exploración de ambientes múltiplemente conectados, de manera distribuida que garantizan la exploración completa del ambiente. Los resultados experimentales muestran que al realizar la exploración en modo cooperativo y aplicando los algoritmos propuestos se obtienen recorridos menores que al realizar la tarea de forma individual.

Palabras Clave: **Exploración distribuida, múltiples robots, comunicación limitada, ambientes desconocidos, obstáculos no distinguibles.**

Abstract of the thesis presented by **Luis Enrique Canul Arceo** as a partial requirement to obtain the Master of Science degree in Master in Computer Science in Computer science.

**Algorithms for exploring unknown environments using multiple robots.**

Abstract approved by:

---

Dr. Edgar Leonel Chávez González

Thesis Co-Director

---

Dr. Ubaldo Ruiz López

Thesis Co-Director

For decades, there has been a lot of interest for robots capable of assisting humans in everyday tasks. To achieve this the robots must be able to navigate through environments autonomously. There are many applications where the help of robots is required to perform exploration tasks, inspection and rescue. The result of the exploration has several applications, among which are the search for a static object and the verification of risks in environments composed by multiple elements. The exploration of environments is a problem that has been studied for many years. There are many works that address the exploration of unknown environments, however, most of them make use of sophisticated sensors and approach the problem by assuming constraints that are often difficult to satisfy in real environments. Most of the proposed solutions do not offer theoretical guarantees that the task will be completed properly. The goal of this work is to guarantee the complete exploration of the environment under certain restrictions. In this work we assume that robots have communication constraints, only have sensors that detect discontinuities and are not able to obtain their position within the environment. Algorithms are proposed for the exploration of multiple connected environments in a distributed way using several robots that guarantee the complete exploration of the environment. The experimental results show that performing the exploration in cooperative mode applying the proposed algorithms results in shorter paths than when the task is performed individually.

**Keywords: Distributed exploration, multiple robots, limited communication, unknown environments, indistinguishable obstacles.**

## Dedicatoria

*A mis padres.*

*A mis hermanos.*

*A mis amigos.*

## Agradecimientos

A Dios, por permitirme vida, salud y fortaleza ante las adversidades.

A mis padres por apoyarme incondicionalmente en las buenas y en las malas, por ser el motor de mi vida, por sus consejos, la educación y el carácter que forjaron en mí, cada éxito es por ellos y para ellos.

A mis asesores, Dr. Edgar Leonel Chávez González y Dr. Ubaldo Ruiz López, por la oportunidad que me dieron de trabajar bajo sus asesorías y por haberme guiado para la culminación de esta tesis, por el apoyo, la dedicación y la paciencia demostrada a lo largo del proceso de realización de este trabajo.

A los miembros del comité, Dra. María del Carmen Maya Sánchez y Dr. Carlos Alberto Brizuela Rodríguez, por el tiempo dedicado en las correcciones y sugerencias para mejorar el trabajo.

A mis compañeros de piso Manuel Combarro y Laura Aneiros, por hacerme sentir parte de su familia, por compartir parte de su cultura conmigo, por aconsejarme y estar siempre pendiente de mí.

A mis amigos, compañeros y gente que conocí durante mi estancia en Ensenada, dentro y fuera del CICESE, por los consejos brindados, la motivación, la confianza depositada en mí y por haber hecho mi estadía más amena.

Al Centro de Investigación Científica y de Educación Superior de Ensenada, por haberme admitido y otorgarme la infraestructura necesaria para llevar a cabo este trabajo de tesis.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico a través de la beca no. 587230 para realizar mis estudios de maestría.

# Tabla de contenido

	Página
Resumen en español . . . . .	ii
Resumen en inglés . . . . .	iii
Dedicatoria . . . . .	iv
Agradecimientos . . . . .	v
Lista de figuras . . . . .	viii
Lista de tablas . . . . .	ix
<b>1. Introducción . . . . .</b>	<b>1</b>
1.1 Antecedentes . . . . .	1
1.2 Planteamiento del problema . . . . .	2
1.3 Propuesta de solución . . . . .	3
1.4 Justificación . . . . .	4
1.5 Objetivos . . . . .	5
1.5.1 Objetivo general . . . . .	5
1.5.2 Objetivos específicos . . . . .	5
1.6 Organización de la tesis . . . . .	5
<b>2. Marco Teórico . . . . .</b>	<b>7</b>
2.1 Visibilidad computacional . . . . .	7
2.1.1 Curva poligonal cerrada . . . . .	7
2.1.2 Polígono . . . . .	7
2.1.3 Visibilidad . . . . .	7
2.1.4 Polígono con agujeros . . . . .	8
2.1.5 Arreglos de segmento de línea . . . . .	8
2.1.6 Conjunto convexo . . . . .	8
2.1.7 Núcleo de un conjunto . . . . .	8
2.1.8 Polígono de visibilidad . . . . .	9
2.1.9 Visibilidad débil y fuerte . . . . .	9
2.1.10 Triangulación . . . . .	9
2.1.11 Grafo de visibilidad . . . . .	9
2.1.12 Línea bitangente . . . . .	10
2.2 Algoritmos insectos . . . . .	10
2.3 Robot móvil . . . . .	11
2.3.1 Holonómicos . . . . .	12
2.3.2 No-holonómicos . . . . .	12
2.4 Cuadrícula de ocupación . . . . .	12
<b>3. Estado del arte . . . . .</b>	<b>14</b>
3.1 Problemas Relacionados . . . . .	14
3.1.1 Problema de exploración . . . . .	14
3.1.2 Problema de cobertura . . . . .	15
3.1.3 Búsqueda de objetos . . . . .	15
3.1.4 Problema <i>rendezvous</i> . . . . .	16
3.2 Precursores del trabajo . . . . .	16
3.2.1 <i>Boundary place graph</i> . . . . .	16
3.2.2 <i>Gap navigation Tree</i> . . . . .	17
3.2.3 Exploración con múltiples robots . . . . .	18
3.3 Características importantes . . . . .	19

<b>4. Algoritmos de exploración propuestos</b> . . . . .	<b>20</b>
4.1 Algoritmos para obstáculos distinguibles . . . . .	20
4.1.1 Sin <i>backtracking</i> . . . . .	20
4.1.2 Con <i>backtracking</i> . . . . .	23
4.1.3 Fusión de las estrategias. . . . .	26
4.2 Algoritmos para obstáculos no distinguibles . . . . .	26
4.3 Completitud de las estrategias de exploración . . . . .	31
<b>5. Experimentos y resultados</b> . . . . .	<b>34</b>
5.1 Diseño de los experimentos . . . . .	35
5.2 Ejecución de los algoritmos . . . . .	35
5.3 Resultados . . . . .	37
<b>6. Conclusiones y trabajo futuro</b> . . . . .	<b>49</b>
6.1 Resumen . . . . .	49
6.2 Conclusiones . . . . .	49
6.3 Trabajo futuro . . . . .	50
<b>Literatura citada</b> . . . . .	<b>52</b>



## Lista de figuras

Figura		Página
1	Tipos de ambientes. . . . .	2
2	Clasificación de obstáculos. Tomado de (Tovar <i>et al.</i> , 2007). . . . .	3
3	Ejemplo de visibilidad en un polígono. Tomado de (Asano <i>et al.</i> , 2000). . . . .	8
4	Grafo de visibilidad. . . . .	10
5	Bitangentes en un conjunto de polígonos. . . . .	10
6	Clasificación de los robots de acuerdo a su aplicación y a las capacidades de desplazamiento. Tomado de (Chacón-Rivas y Garita, 2013). . . . .	12
7	Ejemplo de robot holonómico. Tomado de (Warburton, 2017). . . . .	13
8	Ejemplo de una cuadrícula de ocupación. Tomado de (Duan <i>et al.</i> , 2016). . . . .	13
9	Ambiente de ejemplo. . . . .	21
10	Ejemplo de aplicar la estrategia de exploración al ambiente de la Figura 9. . . . .	24
11	Bitangentes en un ambiente con obstáculos no distinguibles. . . . .	26
12	Ejemplo de dos de las bitangentes generadas por dos triángulos. . . . .	31
13	Proceso de agregar nuevos triángulos de la proposición anterior. (b) al agregar el triángulo $B_k$ , las bitangentes existentes no se ven modificadas. (c) en cambio al agregar $B_l$ una bitangente desaparece, dando lugar a una nueva. . . . .	32
14	Ambientes de prueba. . . . .	34
15	Ejemplos de la ejecución de la exploración. . . . .	36
16	Promedio de distancias, utilizando 9 robots. . . . .	39
17	Promedio de visitas, utilizando 9 robots. . . . .	40
18	Promedio de distancias, utilizando 25 robots. . . . .	41
19	Promedio de visitas, utilizando 25 robots. . . . .	42
20	Promedio de distancias, utilizando 144 robots. . . . .	43
21	Promedio de visitas, utilizando 144 robots. . . . .	44
22	Comparativa de las distintas estrategias en obstáculos distinguibles . . . . .	45
23	Exploración en ambientes no distinguibles. . . . .	46
24	Distancias correspondientes al perímetro y a las bitangentes . . . . .	46
25	Comparativa para los distintos enfoques. . . . .	48

## Lista de tablas

Tabla		Página
1	Precursos del trabajo. . . . .	19
2	Información almacenada en el árbol de la figura 10. . . . .	24
3	Información de los ambientes . . . . .	35
4	Alturas del árbol y promedios de realizar la exploración sin <i>backtracking</i> . . . . .	36
5	Exploración de los ambientes con un robot. . . . .	37
6	Bitangentes utilizadas con un robot. . . . .	47

# Capítulo 1. Introducción

---

## 1.1. Antecedentes

Desde hace años, existe un gran interés por robots autónomos capaces de asistir a la humanidad en tareas de la vida diaria. Para lograrlo, estos deben ser capaces de navegar a través de ambientes de forma autónoma.

La planificación de movimiento, se refiere a la habilidad de un robot para calcular sus movimientos con el fin de alcanzar ciertas metas. Esta se relaciona con el problema desplazamiento del piano (Schwartz y Sharir, 1983) en robótica, la búsqueda de una secuencia de operadores lógicos o de acciones que pasan de un estado inicial a uno final (objetivo) en inteligencia artificial y el diseño de algoritmos que encuentran trayectorias para sistemas no lineales en teoría de control.

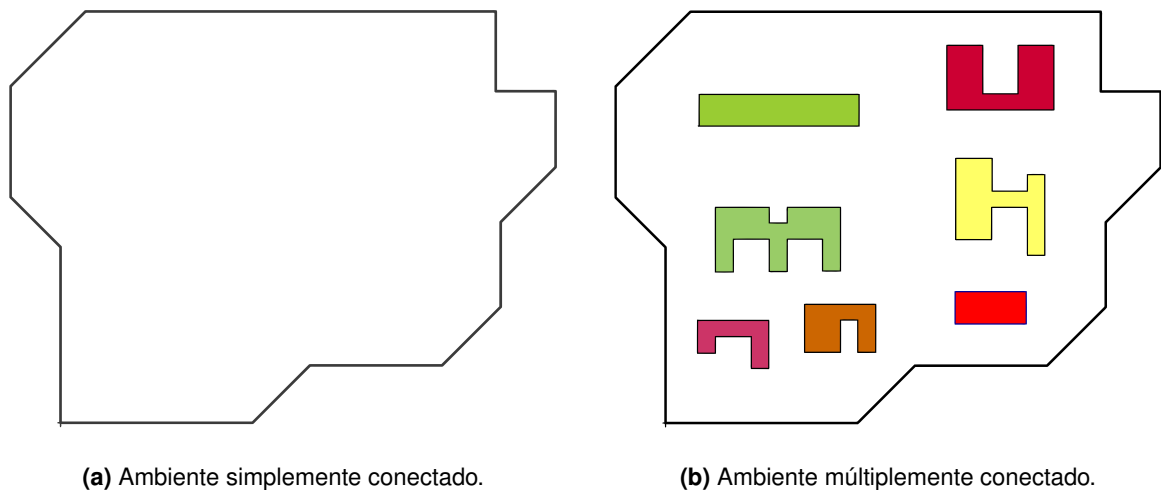
Actualmente, la planificación de movimiento se utiliza en una gran variedad de campos. En robótica, se enfoca en el diseño de algoritmos que generen movimientos útiles para procesar modelos geométricos complicados, dentro de la inteligencia artificial en el diseño de sistemas que usen modelos de decisión para realizar acciones apropiadas, en teoría de control, para calcular trayectorias factibles para sistemas con retroalimentación y optimalidad (Lavalle, 2006).

La planificación de movimientos tiene como finalidad la generación de un conjunto de acciones que permitan a un robot moverse a través de un entorno conocido o desconocido para alcanzar uno o más objetivos partiendo de su posición inicial. Una de las tareas más comunes que se puede resolver mediante la planificación de movimientos es la navegación en entornos con una gran cantidad de obstáculos.

Para llevar a cabo la planificación de movimientos se requiere tener información del ambiente que permita conocer la estructura del mismo. Para recabar esa información el robot hace uso de sensores, los cuales varían en su complejidad así como en la cantidad de información que recaban. Generalmente, mayor información implica una mejor calidad en la planificación del movimiento, sin embargo, tener mayor información viene acompañada de un mayor costo de procesamiento y consumo de energía por parte del robot, por lo tanto, mientras menor sea la cantidad de información que se recabe para realizar adecuadamente una tarea también serán menores los recursos utilizados (Tovar *et al.*, 2007).

En muchos casos, para conocer la estructura del ambiente se requiere realizar una exploración del mismo. La exploración de ambientes junto con la planificación de movimientos es un tema que se ha estudiado durante muchos años. Existen gran variedad de trabajos (Duckett y Nehmzow, 1999; Yao y Gupta, 2011; Kim *et al.*, 2013) que abordan la exploración de ambientes desconocidos, sin embargo, la mayoría utilizan sensores sofisticados y abordan el problema suponiendo gran variedad de restricciones que muchas veces son difíciles de satisfacer en ambientes reales.

Para resolver el problema de exploración, los ambientes se modelan con polígonos y se clasifican en dos tipos: los simplemente conectados y los múltiplemente conectados (Tovar *et al.*, 2007). En la primera clasificación están los polígonos cóncavos y convexos, que no contienen polígonos en su interior. En la segunda clasificación se encuentran aquellos que en su interior cuentan con polígonos ya sean cóncavos o convexos, estos polígonos juegan el papel de obstáculos en el ambiente. Ejemplo de estos ambientes se observan en la Figura 1.

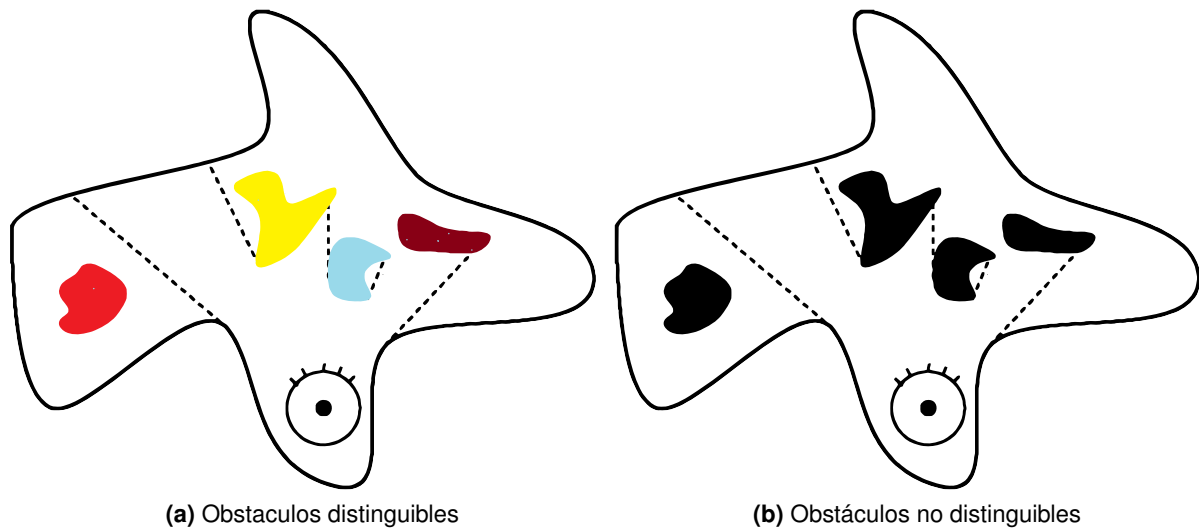


**Figura 1.** Tipos de ambientes.

En los ambientes múltiplemente conectados, se consideran dos casos. En el primero, se supone que los obstáculos son distinguibles entre sí de acuerdo a alguna característica visual, por ejemplo, el color. En el segundo caso, los obstáculos no se pueden distinguir entre ellos. En la Figura 2 se muestra un ejemplo de ambos casos.

## 1.2. Planteamiento del problema

La exploración de ambientes juega un papel muy importante en la robótica móvil. Para realizar la exploración se tienen que tomar en cuenta varios factores como: el tipo de ambiente, el número



**Figura 2.** Clasificación de obstáculos. Tomado de (Tovar *et al.*, 2007).

de robots, los sensores, la comunicación, entre otros.

En la literatura se reportan soluciones al problema de exploración, con uno o múltiples robots, utilizando tecnología y mecanismos sofisticados para compartir información.

En este trabajo, los ambientes se modelan como polígonos y los robots como puntos en el plano. Los robots no tienen acceso a su posición en el ambiente y solo puede detectar discontinuidades en (profundidad) distancia. Estos pueden formar equipos y comunicarse únicamente cuando se encuentren en el mismo lugar. El problema consiste en recorrer el perímetro de todos los obstáculos que contenga el ambiente de manera cooperativa y garantizando que la tarea se realice en un tiempo finito.

El objetivo es desarrollar estrategias que garanticen que la exploración de ambientes múltiplemente conectados se realice completamente.

### 1.3. Propuesta de solución

En este trabajo se aborda el problema de exploración de ambientes desconocidos múltiplemente conectados utilizando múltiples robots, con limitantes de comunicación y con sensores que sólo detectan discontinuidades en profundidad.

Se supone que los robots son capaces de detectar discontinuidades, seguirlas, rodear obstáculos, llevar un registro de los obstáculos explorados y compartir la información con los miembros del equipo, siempre y cuando se encuentren en el mismo lugar.

Para solucionar el problema, se proponen estrategias y algoritmos que garantizan la exploración completa del ambiente. Se descarta utilizar mapas exactos del ambiente e información acerca de la posición de los robots dentro del ambiente y se opta por construir una estructura de árbol que permita viajar a través de los obstáculos utilizando únicamente la información de las discontinuidades en profundidad.

Se proponen tres estrategias para la exploración de ambientes con obstáculos distinguibles y una para los no distinguibles con el propósito de evaluarlas y compararlas entre sí. En la primera, se crea el árbol de la estrategia hasta explorar todos los obstáculos. En la segunda, se crea el árbol hasta alcanzar un nivel dado, se aplica un *backtracking* sobre el árbol, se intercambia la información, se generan nuevos equipos y continua la exploración repitiendo el *backtracking* al alcanzar múltiplos del nivel seleccionado. La tercera estrategia es una fusión de las primeras dos realizando sólo un regreso durante toda la exploración. Para obstáculos no distinguibles, cada robot crea su árbol y deja marcas en los obstáculos disponibles para el resto de los equipos.

Para cada estrategia se forman equipos y se distribuyen los obstáculos por explorar. Al comienzo del algoritmo el equipo se posiciona en un obstáculo, se selecciona un robot del equipo que se encarga de rodearlo partiendo de la posición en la que se encuentra, en busca de obstáculos visibles. Al regresar nuevamente a la posición inicial, comparte la información con el resto del equipo con el fin de crear nuevos equipos y asignarles obstáculos que no se han explorado.

#### **1.4. Justificación**

En la vida real, existen gran variedad de aplicaciones donde se requiere la ayuda de robots para realizar tareas de exploración, por ejemplo, labores de inspección, exploración y rescate, etc. En estos casos los ambientes a explorar son modelados en su mayoría con polígonos múltiplemente conectados. Existen trabajos que realizan estas tareas con ayuda de sensores sofisticados, sin embargo, cada sensor posee error en los datos recabados. Generalmente, se supone que mientras mayor sea la cantidad de información recabada mejor será el desempeño de los algoritmos, sin embargo, al recabar una mayor cantidad de información también implica un mayor error acumulado en la misma. Por esta razón, se pretende minimizar el sensado para reducir los errores de lectura. Además, el tener una menor cantidad de información reduce significativamente el procesamiento en cada robot lo cual se traduce en un menor consumo de energía extendiendo la disponibilidad de los mismos.

Otro punto importante es realizar la exploración del ambiente de modo cooperativo, así se pueden enviar múltiples robots con el objetivo de que el trabajo se realice rápidamente. Esto es importante para aplicaciones donde el tiempo sea un factor decisivo, por ejemplo, en labores de rescate.

El resultado de la exploración tiene variedad de aplicaciones. Entre estas se encuentra la búsqueda de un objeto estático en un ambiente desconocido y la verificación de riesgo en ambientes compuestos por múltiples elementos.

## **1.5. Objetivos**

### **1.5.1. Objetivo general**

Diseñar e implementar algoritmos para la exploración de ambientes desconocidos múltiplemente conectados utilizando múltiples robots que minimicen la información necesaria para realizar sus tareas y garanticen la culminación de las mismas.

### **1.5.2. Objetivos específicos**

- Analizar y estudiar las técnicas en el estado del arte.
- Caracterizar las soluciones existentes.
- Diseñar el algoritmo para el problema planteado.
- Establecer estrategias para un trabajo cooperativo entre los robots.
- Implementar el algoritmo en simulación.
- Realizar la comparativa entre la exploración con uno y múltiples robots.
- Realizar pruebas de las estrategias en ambientes controlados.

## **1.6. Organización de la tesis**

La tesis se divide en 6 capítulos, organizados de la siguiente manera: En el capítulo 2, se presentan algunos conceptos importantes para una mejor comprensión del trabajo. En el capítulo 3, se discuten y se comparan los trabajos que dan pauta al desarrollo de esta tesis. En el capítulo

4, se describen los algoritmos que se proponen para la solución al problema planteado. Posteriormente se presentan los experimentos y los resultados de aplicar los algoritmos propuestos, todo esto en el capítulo 5. Finalmente, en el capítulo 6, se presentan las conclusiones obtenidas y también las propuestas para trabajos futuros que surgieron con base a este trabajo.



## Capítulo 2. Marco Teórico

---

En este capítulo, se describen conceptos básicos utilizados en el área de planeación de movimiento que servirán para formalizar el problema de exploración de ambientes. Se destacan temas de geometría computacional tomados principalmente de los autores (Asano *et al.*, 2000; de Berg *et al.*, 2008) y algoritmos insecto de (Choset, 2005).

### 2.1. Visibilidad computacional

Para definir visibilidad se requiere de ciertos conceptos los cuales se mencionan a continuación.

#### 2.1.1. Curva poligonal cerrada

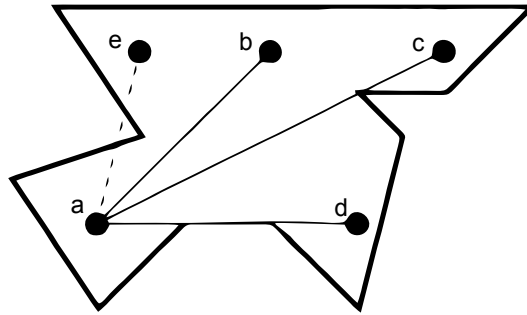
Una curva poligonal cerrada es una curva que se compone de un conjunto de vértices de al menos tres elementos,  $V = \{v_1, v_2, \dots, v_n\}$ , tal que  $v_i \in \mathbb{R}^2$  y por un conjunto de aristas  $E = \{e_1, e_2, \dots, e_n\}$  donde cada  $e_i$  es un segmento de línea con extremos que van de  $v_i$  a  $v_{i+1}$ , además es necesario que se intersecten pares de aristas, la intersección de estas debe ser de la forma  $e_i, e_{i+1}$ .

#### 2.1.2. Polígono

Un polígono es una curva poligonal cerrada junto con su área. El vértice de un polígono es llamado convexo si el ángulo interior entre dos vértices incidentes es menor a  $\pi$ , de otra manera es llamado cóncavo.

#### 2.1.3. Visibilidad

Se define la visibilidad dentro del plano cartesiano, se dice que  $x$  y  $y$  son visibles en un polígono, si el segmento  $xy$  no contiene puntos del exterior del polígono en cuestión, esto se puede ver en la Figura 3, se aprecia que  $a$  y  $e$  no son visibles. Esta definición permite que todos los segmentos de visibilidad pasen por algún vértice cóncavo o por una arista. También se puede decir que  $x$  ve a  $y$  si  $x$  y  $y$  son visibles.



**Figura 3.** Ejemplo de visibilidad en un polígono. Tomado de (Asano *et al.*, 2000).

#### 2.1.4. Polígono con agujeros

El polígono con agujeros, es aquel al que se le han removido subpolígonos, con este se puede modelar un cuarto sin luz con varios muebles, al encender una luz, ciertas áreas no serán parte del polígono.

#### 2.1.5. Arreglos de segmento de línea

Un arreglo de segmentos de línea es una colección de segmentos en el plano. Esta colección puede ser el contorno de un polígono o de un polígono con agujeros. En estos arreglos también es posible definir la visibilidad. Dos puntos  $x$  y  $y$  en el plano son visibles si el segmento  $xy$  no interseca con algún elemento del arreglo.

#### 2.1.6. Conjunto convexo

Un conjunto de puntos (por ejemplo, un polígono) es llamado convexo si todos los puntos son visibles, el conjunto  $S$  es convexo si  $\forall x, y \in S$   $x$  ve a  $y$ .

#### 2.1.7. Núcleo de un conjunto

El núcleo de un conjunto  $S$ ,  $K(S)$  es el conjunto de todos los puntos en  $S$  que son visibles entre si. Formalmente se define como  $K(S) = \{x \in S \mid \forall y \in S$   $x$  ve a  $y\}$ . Es evidente que un conjunto convexo  $S$  tiene  $K(S) = S$ . El núcleo de un conjunto también se relaciona con conjuntos en forma de estrella, el cual es el núcleo de un conjunto  $K(S)$  con la condición que  $K(S) \neq \emptyset$ .

### 2.1.8. Polígono de visibilidad

El polígono de visibilidad de un punto  $p$  en un polígono  $P$  denotado por  $V(p)$ , es el conjunto de todos los puntos que son visibles por  $p$ . Formalmente,  $V(p) = \{q \in P \mid p \text{ ve a } q\}$ , esta definición es válida también para polígonos con agujeros y para arreglos de segmentos de línea.

### 2.1.9. Visibilidad débil y fuerte

A continuación, se presentan los conceptos de visibilidad débil y visibilidad fuerte. Sea  $S$  un subconjunto de un polígono  $P$ , un punto  $p$  es débilmente visible de  $S$  si este es visible para algún punto del conjunto, al conjunto de todos esos puntos se le conoce como polígono de visibilidad débil de  $S$  y se denota como  $V(S)$ . De manera similar, se dice que un punto del subconjunto  $S$  es completamente visible si para todos los puntos este es visible, al conjunto de esos puntos se le conoce como polígono de visibilidad completa de  $S$ .

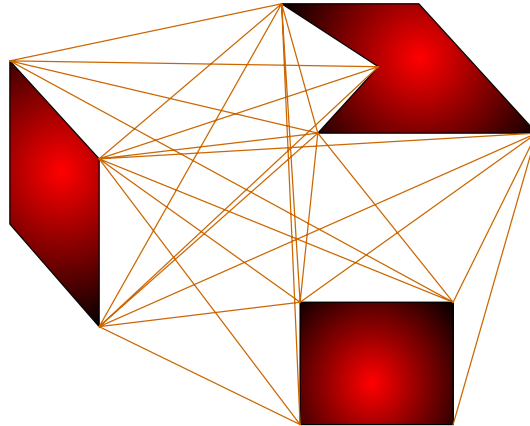
### 2.1.10. Triangulación

Una diagonal de un polígono es un segmento de línea, que tiene por extremos pares de vértices visibles. Una triangulación es la división en triángulos de un polígono a partir de sus diagonales. Aunque la definición implica que la división únicamente tome las diagonales, también es posible hacer una triangulación a partir de cualquier segmento, esto da como resultado puntos de intersección, los cuales se conocen como puntos de Steiner, a este tipo de triangulación se le conoce como triangulación con puntos de Steiner. En un polígono con triangulación, se habla de oreja si el triángulo está formado por los vértices consecutivos  $v_i, v_{i+1}$  y  $v_{i+2}$ . Se puede utilizar un árbol para representar la triangulación, el árbol de una triangulación es el que tiene un nodo por cada triángulo resultante de la división y un arco entre los pares de nodos que comparten la misma diagonal.

### 2.1.11. Grafo de visibilidad

Un grafo de visibilidad de un polígono  $P$ , denotado por  $VG(P)$  es un grafo que muestra la relación de todos los vértices que son visibles en  $P$ . Dado un conjunto de obstáculos con forma poligonal en el plano cartesiano se dice que el grafo de visibilidad es aquel grafo en el cual cada nodo representa un vértice de los polígonos y las aristas son las conexiones visibles entre tales vértices. Esto quiere decir que para cada arista en el grafo de visibilidad definida por dos vértices,

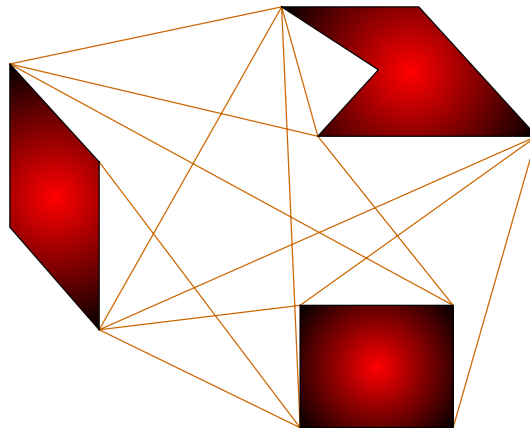
el segmento de recta que conecta los vértices correspondientes en el plano no se intersecta con ningún polígono. Ejemplo de lo anterior se visualiza en la Figura 4. Un grafo de visibilidad se define similarmente en polígonos con agujeros y en arreglos de segmentos de línea.



**Figura 4.** Grafo de visibilidad.

### 2.1.12. Línea bitangente

Una bitangente es una línea que es tangente a una curva cerrada en dos puntos distintos. Si tenemos un conjunto de polígonos en el plano cartesiano, se puede decir que una bitangente es una tangente que es común a dos vértices. En la Figura 5, se pueden observar todas las bitangentes que son generadas por el conjunto de polígonos.



**Figura 5.** Bitangentes en un conjunto de polígonos.

## 2.2. Algoritmos insectos

Los algoritmos insecto son utilizados para la navegación de robots móviles en ambientes desconocidos, cuando se busca minimizar la cantidad de información requerida para efectuar la tarea. Estos algoritmos almacenan una mínima cantidad de puntos y no generan un mapa completo del

ambiente. Suponen que el objetivo es conocido pero los obstáculos desconocidos (Buniyamin *et al.*, 2014). Los algoritmos insectos poseen características idóneas para nuestro problema, por lo tanto, se pretende aprovecharlas en el desarrollo de nuestra solución.

Los algoritmos insecto pueden ser implementados como planificadores simples para la construcción de rutas de navegación en ambientes reales con presencia de obstáculos. En general funcionan de la siguiente manera: se supone que se sabe cuál es el objetivo y se fija la dirección en la que el robot debe moverse para rodear los obstáculos cuando se presentan colisiones durante la exploración. El robot se dirige hacia el objetivo, si se detecta algún obstáculo dentro de la trayectoria este lo rodea y guarda el punto que minimiza la distancia hacia el objetivo, una vez que el obstáculo ha sido rodeado completamente, el robot regresa al punto para continuar su camino hacia el objetivo.

Los algoritmos insecto actúan bajo tres hipótesis acerca del robot móvil:

- i) El robot es un punto.
- ii) Maneja localización perfecta (no incertidumbre).
- iii) Sensores precisos.

El conjunto de algoritmos insecto se compone de los siguientes: Bug1 , Bug2 y DistBug, entre otros (Márquez *et al.*, 2011).

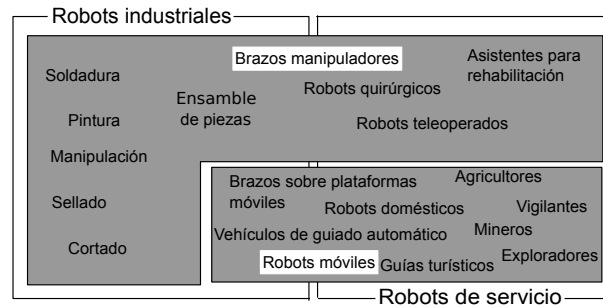
### **2.3. Robot móvil**

Un robot es una entidad virtual o mecánica artificial que normalmente es conducido por un programa de una computadora o por un circuito eléctrico, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio.

Entre las aplicaciones de los robots, se puede observar que en la industria los utilizan como herramientas que sustituyen a los recursos humanos especialmente los robots manipuladores. Pero también se utilizan robots móviles como vehículos guiados automatizados.

Se clasifican los robots de acuerdo a sus aplicaciones como industriales o de servicio y de acuerdo a sus capacidades de desplazamiento, entre los que se encuentran los brazos manipuladores y los robots móviles (Chacón-Rivas y Garita, 2013). Se puede observar en la Figura 6

algunos ejemplos que caen en dicha clasificación. A su vez los robots móviles se clasifican en holonómicos y no-holonómicos por la manera en la que se mueven.



**Figura 6.** Clasificación de los robots de acuerdo a su aplicación y a las capacidades de desplazamiento. Tomado de (Chacón-Rivas y Garita, 2013).

### 2.3.1. Holonómicos

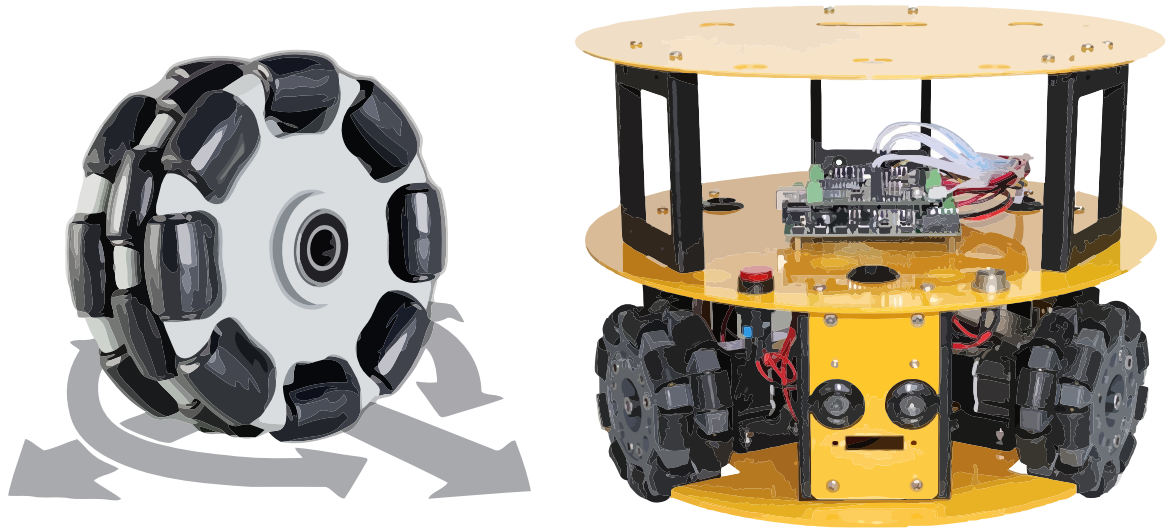
Holonómico se refiere a la relación entre los grados de libertad controlables y el total de grados de libertad del robot. Si los grados de libertad que son controlables es igual al total de grados de libertad, entonces se dice que el robot es holonómico. Un robot construido sobre ruedas giratorias o ruedas omnidireccionales es un buen ejemplo de un robot holonómico en virtud de que puede moverse libremente hacia cualquier dirección, nótese que los grados de libertad controlables son iguales a los grados de libertad totales que posee. La Figura 7, en (a) se muestra una rueda giratoria la cual puede rotar en cualquier dirección y en (b) un ejemplo de un robot holonómico.

### 2.3.2. No-holonómicos

Si los grados de libertad que son controlables es menor al total de los grados de libertad, entonces se dice que es un robot no-holonómico. Un robot similar a un carro cuenta con tres grados de libertad; la posición en dos ejes y su orientación. Sin embargo, únicamente dos son los controlables, la aceleración y los ángulos de vuelta de las ruedas, lo que provoca que sea difícil que el robot gire a cualquier dirección.

## 2.4. Cuadrícula de ocupación

En la robótica móvil muchas veces es necesario crear un mapa del ambiente y definir métodos para localizar el robot dentro de él. El mapeo es el problema de determinar el mapa correspondiente al ambiente, mientras que la localización es el problema de encontrar la posición del robot.

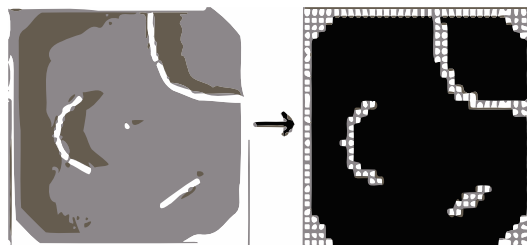


(a) Rueda giratoria

(b) Robot holonómico

**Figura 7.** Ejemplo de robot holonómico. Tomado de (Warburton, 2017).

Una característica a notar es que muchas técnicas de localización dependen en esencia de la representación del mapa. Una muy común es el uso de cuadrículas de ocupación (Milstein, 2008) para aproximar el ambiente. Una cuadrícula de ocupación representa el ambiente como bloques de celdas. Se distinguen dos estados para cada celda: ocupado y desocupado. Cuando se habla de una celda ocupada se entiende que el robot no puede atravesarla, únicamente cuando la celda este desocupada puede ser atravesada. En la Figura 8, se aprecia un ejemplo de una cuadrícula de ocupación.

**Figura 8.** Ejemplo de una cuadrícula de ocupación. Tomado de (Duan *et al.*, 2016).

La calidad de la aproximación del ambiente depende del tamaño de las celdas, mientras más pequeñas sean las celdas, más exacta será la representación del ambiente.

## Capítulo 3. Estado del arte

---

### 3.1. Problemas Relacionados

#### 3.1.1. Problema de exploración

Desde los comienzos de la robótica móvil el problema de exploración ha sido fundamental en el área. El objetivo de un proceso de exploración es cubrir totalmente el ambiente en un tiempo finito. En (Burgard *et al.*, 2005) el problema se resuelve utilizando múltiples robots, la justificación del por qué se aborda de esta manera radica en las ventajas que ofrece el utilizar varios robots. Por ejemplo, un conjunto robots, ofrece mayores posibilidades de acabar en menor tiempo del que requiere un solo robot.

Bajo este enfoque los robots necesitan conservar el registro de las áreas que han sido exploradas, crear un mapa global que les permita seguir un camino y coordinar sus acciones. Esto es posible porque a medida que se crea el mapa, se comparte con otros robots y estos a su vez comparten las posiciones que ocupan dentro del ambiente.

El mapa que se crea típicamente se representa como una cuadrícula de ocupación, sin embargo, puede extenderse a un mapa topológico o un mapa de cobertura. A lo largo de esta sección, la representación utilizada será la cuadrícula. Cada celda de la misma contiene un valor numérico que representa la probabilidad de que dicha celda cubra un obstáculo.

Los mapas generalmente contienen celdas desconocidas hasta cierto instante, a medida que el robot avanza se actualiza a un valor conocido, por lo tanto la exploración es un concepto binario, entre áreas conocidas y desconocidas.

Para las labores de exploración es indispensable identificar las áreas conocidas de las no conocidas para luego calcular el costo del recorrido de cada robot. El interés radica en las celdas límite, es decir, aquellas a las cuales se les conoce su valor y tienen como vecina a al menos a una celda con valor desconocido. Se espera que si se dirige el robot a las celdas límite, al llegar, este habrá reunido información de las celdas por explorar. Para evitar que varios robots se dirijan al mismo lugar, se necesita establecer ubicaciones objetivo.

Se utiliza un enfoque de decisión teórico para determinar las ubicaciones objetivo apropiadas para conseguir una exploración eficiente. Para ello se establece un costo para alcanzar una cel-



da, y la utilidad que esta posee. Con estos parámetros y con programación dinámica es posible realizar la exploración completa del ambiente.

### 3.1.2. Problema de cobertura

La cobertura es un paradigma para evaluar sistemas robóticos, se definen tres tipos de cobertura básicos, cobertura general, cobertura de barrera, cobertura de barrido (Gage, 1992).

- **Cobertura general:** El objetivo es alcanzar un arreglo estático de elementos que maximicen el total del área detectada.
- **Cobertura de barrera:** El objetivo es minimizar la probabilidad de no detectar la penetración de la barrera.
- **Cobertura de barrido:** El objetivo es mover un grupo de elementos a través de un área de cobertura, de manera que se alcance un balance entre maximizar el número de detecciones por tiempo y minimizar el número de detecciones perdidas por área.

El problema de la cobertura general se ha estudiado ampliamente en robótica móvil para las tareas de navegación libre de colisiones. El desarrollo del problema es similar a la exploración con múltiples robots y al problema de la creación de mapas. El problema de cobertura ha sido estudiado por varios autores (Cortés *et al.*, 2004; Iyengar, 2007; Misra *et al.*, 2009; Liang y Chen, 2011).

Finalmente, el problema de cobertura general está estrechamente relacionado al problema de la galería de arte en geometría computacional. En el problema anterior, se busca determinar el número mínimo de cámaras necesarias para garantizar que un ambiente poligonal es vigilado completamente (O'Rourke, 1987).

### 3.1.3. Búsqueda de objetos

Otro problema relacionado con la exploración de ambientes desconocidos es la búsqueda de uno o más objetos reconocibles (Taylor y Kriegman, 1995; Sarmiento *et al.*, 2004; Tovar *et al.*, 2007; Veiga *et al.*, 2016). Este problema, es equivalente al problema de encontrar todos los objetos en un ambiente, ya que el robot puede tener que descubrir todos los objetos antes de decidir si el objetivo está presente o no. Existen muchas aplicaciones donde se requiere realizar la búsqueda

de objetos en un ambiente desconocido, desde tareas sencillas como subir todas las cajas de un mismo color a un camión, inspeccionar un área en busca de fugas en barriles de residuos nucleares, hasta labores domésticas (Veiga *et al.*, 2016).

Para llevar a cabo la búsqueda de los objetos, el robot está equipado con un sistema de reconocimiento visual, con el que realiza una exploración sistemática del entorno en busca de uno o más objetos reconocibles.

#### **3.1.4. Problema *rendezvous***

El problema de *rendezvous* se refiere al desafío algorítmico para que dos o más entidades móviles se reúnan en el mismo tiempo y punto del espacio. El problema de reunión puede ser una tarea por sí sola o puede formar parte de un proceso de comunicación o coordinación más complejo en el que los robots estén involucrados. El espacio puede ser una red de nodos discretos entre los cuales los robots puedan moverse a lo largo de las conexiones existentes o un entorno geométrico en el que el movimiento de los robots sólo esté restringido por las propiedades topológicas del espacio. Con el fin de realizar la reunión, los robots deben acordar un mecanismo de encuentro. La viabilidad y eficiencia de la solución adoptada depende de la capacidad de los robots para moverse, observar y comunicarse (Gąsieniec, 2013).

Al igual que los problemas anteriores existen distintas formas de abordar el problema, dependiendo del tipo y restricciones de comunicación que los robots posean (Park y Hutchinson, 2016).

### **3.2. Precursores del trabajo**

#### **3.2.1. *Boundary place graph***

El problema de explorar un ambiente desconocido en busca de uno o más objetos reconocibles se considera en (Taylor y Kriegman, 1995), donde se suponen capacidades limitadas de detección por parte del robot. El ambiente se representa con una estructura denominada *boundary place graph* en el que se lleva el registro de puntos de referencia. Un punto de referencia es simplemente un objeto distintivo o patrón, que el robot puede reconocer con su sistema de visión siempre que esté a la vista. El *boundary place graph* registra el conjunto de puntos de referencia que son visibles desde los límites de varios obstáculos dentro del ambiente. Esta estructura no registra ninguna representación de la geometría del entorno ni almacena prescripciones explícitas (trayectorias) para moverse entre lugares.

La estructura se va creando a medida que avanza la exploración. Una vez completada la exploración, se puede utilizar el resultado para planear o ejecutar futuras tareas de navegación dentro del ambiente. Algunos aspectos importantes a resaltar son: no se supone ningún conocimiento previo sobre la estructura geométrica del entorno, no se proporciona ninguna información previa sobre las posiciones absolutas o relativas de los obstáculos, no se intenta medir distancias. Tampoco se supone conocimiento sobre el número de obstáculos que se podría encontrar.

Se tiene como premisas que la exploración se realiza en un ambiente múltiplemente conectado. El robot es capaz de diferenciar cada uno de los objetos, entre opacos y transparentes. El robot se representa por medio de un punto en el plano y se supone que es holonómico.

El robot además de detectar los objetos con un sistema de reconocimiento visual, puede dirigirse a ellos y rodearlos. A medida que se desplaza el robot va creando la estructura. Cuando se encuentra un obstáculo, el robot se dirige a él, lo rodea y agrega un elemento a la estructura.

En esta tesis se supone que el robot cuenta con características similares a este enfoque, sin embargo, las diferencias están en el sensor que utiliza el robot, la representación de la exploración, la caracterización de los obstáculos y el número de robots que utiliza para la exploración.

### **3.2.2. *Gap navigation Tree***

El trabajo presentado en (Tovar *et al.*, 2007) propone el uso de una estructura denominada *Gap Navigation Tree* (GNT) para la navegación con un robot, sin utilizar coordenadas.

El robot se modela como un punto que se mueve dentro del ambiente. Se supone que el ambiente es un polígono simplemente conectado en el plano cartesiano. El sensor que posee el robot es llamado *gap sensor* y obtiene información de las discontinuidades en profundidad, el robot no tiene información alguna de la posición dentro del ambiente.

La información de las discontinuidades es fundamental para la creación del mapa y consecuentemente la planificación del movimiento dentro del ambiente. Cuando el robot detecta una discontinuidad, se dirige hacia ella y puede seguirla (rodearla), detenerse en ella o seguir de largo. La información que recaba el sensor se utiliza para crear la lista de discontinuidades para poder crear la estructura (*Gap Navigation Tree*). Un aspecto que se resalta es que bajo estas condiciones se demuestra que la exploración es completa y óptima.

El enfoque anterior se extiende en (Lopez-Padilla *et al.*, 2013), a un robot en forma de disco,

la importancia de la forma del robot radica en que muchas situaciones de la vida real necesitan modelos de robots complejos. El sensor para este enfoque detecta discontinuidades de dos tipos, *near to far* y *far to near*, es importante mencionar que esta clasificación es posible porque el robot puede calcular la distancia hacia la discontinuidad detectada. El robot se va desplazando a través del ambiente basándose en la información recabada y de las políticas de movimiento que posee, rotar en el mismo sitio (derecha o izquierda), seguir una línea recta hacia otro objetivo, girar en una discontinuidad (derecha o izquierda). Con las condiciones establecidas, se logra una exploración completa y se garantiza que sea óptima.

### 3.2.3. Exploración con múltiples robots

Algunas estrategias de exploración utilizan la exploración basada en la frontera y múltiples robots. En esencia cuando se trabaja con la frontera, el robot se dirige a la línea imaginaria que divide las partes conocidas y desconocidas del ambiente (Yamauchi, 1997).

Muchos trabajos abordan el problema de exploración utilizando múltiples robots (Burgard *et al.*, 2005; Wurm *et al.*, 2008). En (Burgard *et al.*, 2005), el mapa se representa utilizando una cuadrícula de ocupación y las posibles ubicaciones para el siguiente paso de exploración se definen sobre las celdas situadas en el borde entre las celdas conocidas y desconocidas. En (Wurm *et al.*, 2008), los autores proponen una estrategia de exploración multi-robot en la que en lugar de fronteras, los autores utilizan una segmentación del ambiente para determinar objetivos de exploración para robots individuales. Esta segmentación mejora la distribución de los robots dentro del ambiente. En (Matignon *et al.*, 2012), los autores proponen un método para la exploración con múltiples robots basado en el proceso de decisión descentralizado de Markov.

En (Lee *et al.*, 2016), los autores proponen algoritmos distribuidos para la construcción de una triangulación con ayuda de múltiples robots. Los enfoques que se aplican son exploración, cobertura y vigilancia, utilizando un conjunto de robots con capacidades limitadas. Algunas diferencias importantes entre este trabajo de tesis y el de (Lee *et al.*, 2016) son los siguientes: en este trabajo, se consideran varios entornos múltiplemente conectados mientras que en (Lee *et al.*, 2016) los autores consideran entornos simplemente conectados. En este trabajo, los robots siguen los límites de los obstáculos, esto hace que este trabajo se relacione de alguna manera con los algoritmos insectos.

En (Ng y Bräunl, 2007), los autores comparan el rendimiento de varios algoritmos insectos en

**Tabla 1.** Precursores del trabajo.

Representación de la estrategia	Autor (es)	Número de robots	Tipo de sensor	Información	Tipo de ambiente
Boundary place graph	(Taylor y Kriegman, 1995)	Uno	Sistema de visión	Puntos de referencia	Múltiplemente conectado
Gap Navigation Tree	(Tovar <i>et al.</i> , 2007)	Uno	Gap sensor	Discontinuidades en profundidad	Simplemente conectado
	(Lopez-Padilla <i>et al.</i> , 2013)	Uno	Gap sensor	Discontinuidades en profundidad	Simplemente conectado
Cuadrícula de ocupación	(Burgard <i>et al.</i> , 2005)	Múltiples	Sofisticados, heterogéneos	Celdas conocidas, desconocidas	Ambos
	(Maignon <i>et al.</i> , 2012)	Múltiples	Sensor laser y odométrico	Celdas conocidas, desconocidas	Ambos
Segmentación de ambientes	(Wurm <i>et al.</i> , 2008)	Múltiples	Sofisticados, heterogéneos	Objetivos de exploración	Ambos
	(Lee <i>et al.</i> , 2016)	Múltiples	Sensor con capacidades	Triangulaciones	Simplemente conectado
Sin representación	(Ng y Bräunl, 2007)	Uno	Sensor de rango	Objetivo y sus coordenadas	Ambos

términos de diferentes criterios, por ejemplo, la distancia recorrida para alcanzar una meta. Sin embargo, no analizan el rendimiento de los sistemas con múltiples robots para la tarea de cubrir el ambiente. Tampoco se estudia el efecto de la comunicación limitada entre los robots y cómo afecta en la distancia recorrida por los robots.

### 3.3. Características importantes

En la Tabla 1 se presentan algunas características de los trabajos que dieron pauta a esta tesis. Se observa que para ambientes simplemente conectados se utiliza una representación basada en árboles y con garantías teóricas de una exploración completa, sin embargo, para ambientes múltiplemente conectados aún no es posible. Los sensores más simples se utilizan en ambiente simplemente conectados por lo que el reto es poder utilizar dichos sensores en ambientes múltiplemente conectados.

Los trabajos mostrados en la Tabla 1 forman la base de este trabajo de tesis ya que se combinan características como la representación, el tipo de sensor, la cantidad de robots a utilizar, el ambiente a explorar, entre otras.

## Capítulo 4. Algoritmos de exploración propuestos

---

A continuación, se presentan los algoritmos propuestos para la exploración de ambientes múltiplemente conectados. Se consideran dos enfoques, en el primero los obstáculos son identificados de manera única mientras que en el segundo no.

### 4.1. Algoritmos para obstáculos distinguibles

En este enfoque se establece que cada obstáculo es identificable de manera única, esto puede interpretarse como que cada obstáculo tiene una característica visual distintiva, por ejemplo, su color. También se supone que los robots se pueden comunicar con los demás sólo si se encuentran en el mismo punto del obstáculo.

Se recalca que un obstáculo se considera explorado si se visitan sus vértices y se almacena las bitangentes que parten de él. Se dice que la exploración es completa cuando todos los obstáculos fueron explorados al menos una vez.

El objetivo es explorar todos los obstáculos utilizando múltiples robots. Para comenzar la exploración, se crea un equipo con todos los robots y se posicionan en un obstáculo. Un robot del equipo se encarga de rodear el obstáculo y almacenar las bitangentes que detecta, que equivale a la información que recaba el robot.

El robot que rodea el obstáculo es el encargado de compartir la información con el resto del equipo, con la finalidad de formar nuevos equipos y continuar la exploración de todo el ambiente.

El problema se aborda de tres formas distintas, en la primera se realiza la exploración y se genera una estructura de árbol que contiene la información relacionada con los obstáculos explorados, en la segunda se realiza *backtracking* sobre la estructura para compartir información y en la tercera se combinan las estrategias anteriores.

#### 4.1.1. Sin *backtracking*

Supongamos que hay  $n$  robots y  $m$  obstáculos distinguibles en  $E$ . Cada robot se denota como  $r_i$  donde  $i = 1, \dots, n$  y cada obstáculo como  $O_j$  tal que  $j = 1, \dots, m$ , en la Figura 9 se muestra un ejemplo de un ambiente con 7 obstáculos. Inicialmente todos los robots se encuentran en el vértice de un obstáculo y desde esa posición calculan la secuencia de bitangentes detectadas en

orden angular, en otras palabras significa encontrar la secuencia de obstáculos que son visibles desde ese vértice. Cada robot conserva el registro de los obstáculos detectados y los almacena en un cola  $D_i$ . Es probable que se detecten varias bitangentes hacia el mismo obstáculo, en ese caso solo la primera se toma en cuenta. El procedimiento descrito se puede ver en el Algoritmo 1.

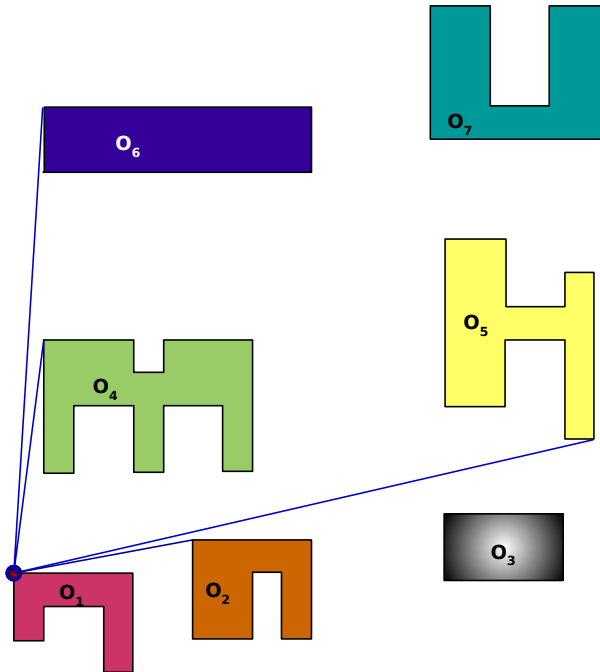


Figura 9. Ambiente de ejemplo.

---

**Algoritmo 1** Inicio\_exploración\_obstáculo\_distinguible

---

**Entrada:** Un ambiente desconocido  $E$ , un equipo  $R$  de  $n$  robots inicialmente posicionados en un vértice de un obstáculo.

**Salida:** Árbol de la estrategia

Se declara  $D_i$  donde  $i = 1, \dots, n$

Se fija la dirección para rodear el obstáculo

Se define  $O_s$  como el obstáculo en el que se encuentra el robot

**para**  $i = 0$  to  $n$  **hacer**

    Agregar a la lista  $D_i$  los obstáculos detectados por los robots

**fin para**

Se declara  $T$  que es la representación de la estrategia

Exploración\_obstáculo\_distinguible( $E, R, O_s, T, \text{NIL}$ )

**regresar**  $T$

---

El Algoritmo 1 llama al Algoritmo 2. Se selecciona un robot  $r_s$  del equipo, el cual tiene la característica de poseer el identificador más pequeño. Este rodea el obstáculo  $O_t$  en el que se encuentra siguiendo una dirección fijada al inicio del algoritmo.

---

**Algoritmo 2** Exploración\_obstáculo\_distinguible
 

---

**Entrada:** Un ambiente desconocido  $E$ , un equipo  $R$  de  $m$  robots, un obstáculo  $O_s$  que se explorará, el árbol  $T$  de la estrategia, nodo actual  $n_p$  en  $T$ .

Se define  $\partial O_s$  como el borde de  $O_s$

Seleccionar un robot  $r_s$  de  $R$

Marcar con  $p_s$  un punto de  $O_s$

**repetir**

$r_s$  sigue  $\partial O_s$  y actualiza su  $D_s$  con los nuevos obstáculos detectados

**hasta que** Encuentre  $p_s$

Compartir el  $D_s$  con el resto del equipo

Crear el nodo  $n_t = (O_s, R, D_s)$

**si**  $n_p \neq \text{NIL}$  **entonces**

Agregar  $n_t$  como hijo de  $n_p$

**si no**

$n_t$  se agrega como raíz del árbol

**fin si**

Declarar  $|D_s^{ne}|$  como el número de obstáculos no visitados en  $D_s$

**si**  $|D_s^{ne}| \neq 0$  **entonces**

**si**  $|D_s^{ne}| < n$  **entonces**

Crear una cola  $Q$  con los  $D_s$ .

**para**  $i = 0$  to  $m$  **hacer**

Asignar visitado a todos obstáculos no explorados de  $D_i$ .

**fin para**

Crear  $R'_i$  tal que  $i = 1, \dots, |D_s^{ne}|$

Distribuir los  $m$  robots en los equipos.

**para**  $i = 0$  to  $|D_s^{ne}|$  **hacer**

Sacar un obstáculo  $O_t$  de  $Q$

Mover el equipo  $R'_i$  a  $\partial O_t$

Exploración\_obstáculo\_distinguible( $E, R'_i, O_t, T, n_p$ )

**fin para**

**si no**

Crear la cola  $Q$  con los primeros  $m$  obstáculos no explorados en  $D_s$

**para todo**  $q \in Q$  **hacer**

**para**  $i = 0$  to  $m$  **hacer**

Asignar  $q$  en  $D_i$  como visitado

**fin para**

**fin para**

Crear  $R'_i$  tal que  $i = 1, \dots, m$

**para**  $i = 0$  to  $|D_s^{ne}|$  **hacer**

Sacar un obstáculo  $O_t$  de  $Q$

Mover el equipo  $R'_i$  a  $\partial O_t$

Exploración\_obstáculo\_distinguible( $E, R'_i, O_t, T, n_p$ )

**fin para**

**fin si**

**fin si**

---

Un robot es capaz de dirigirse hacia la discontinuidad detectada, sin embargo, debido a que las discontinuidades dependen del vértice en el que se encuentre el robot, al llegar a la discon-



tinuidad, detectará discontinuidades distintas a la que lo dirigió a su posición. Al desplazarse en cualquier otra dirección, el robot no sabrá cual fue el punto al que llegó al obstáculo. Para tratar este problema, al llegar el robot coloca una marca que indica el punto del primer contacto entre el robot y el obstáculo. En nuestro caso, debido a que el robot rodea el obstáculo, la marca tiene el propósito de indicar que se ha rodeado el obstáculo siempre y cuando se encuentre la marca que se colocó al llegar a este.

El robot  $r_s$  se sitúa en el marcador de  $O_t$ , lo rodea y actualiza la cola  $D_s$  con los nuevos obstáculos detectados. Al encontrar nuevamente el marcador se entiende que el obstáculo ha sido explorado, por lo que se marca como  $O_t^e$ . Al estar nuevamente en la posición del equipo toma el marcador y comparte la  $D_s$  con el resto de los miembros, por lo tanto  $D_i = D_s$ . A partir de las  $D_i$ 's se calcula el número de obstáculos no explorados en  $D_i$  y se denota con  $|D_i^{ne}|$ .

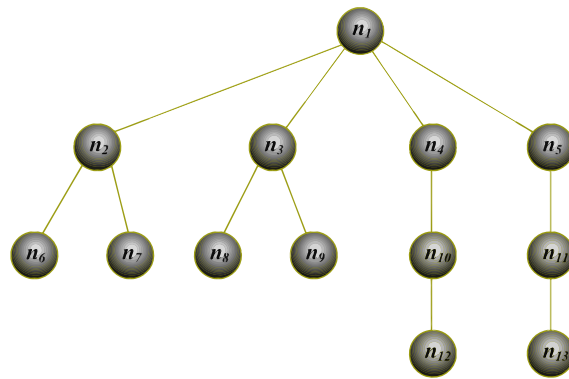
Para visitar los obstáculos en  $D_i$ , se utiliza la siguiente estrategia. Si  $|D_i^{ne}| < n$  entonces  $k \in \mathbb{N}$  robots se asigna a cada elemento del  $D_i^{ne}$ , donde  $n = k|D_i^{ne}| + l$  tal que  $l < |D_i^{ne}|$ . Los  $l$  robots restantes son también distribuidos equitativamente, se puede notar que cada obstáculo no visitado en  $D_i$  es visitado a lo más por  $k + 1$  robots. Por otra parte si  $|D_i^{ne}| > n$ , cada robot es asignado a un obstáculo y en este caso se inicia una exploración independiente en  $E$ .

Antes de partir del obstáculo, todos los robots marcan como visitado los obstáculos que se visitarán en su  $D_i$  con la finalidad de evitar que los asignados se visiten nuevamente. Cuando el equipo alcance al obstáculo asignado, se realiza nuevamente el procedimiento anterior hasta que todos los elementos de las  $D_i$  estén marcados como visitados.

La estrategia antes descrita se puede representar por medio de un árbol, donde los nodos almacenan información del obstáculo visitado, los miembros del equipo que lo visitó, y las  $D_i$  actualizadas después de explorarlo. Los hijos de los nodos representan los obstáculos no explorados que se han detectado desde el nodo padre. Un ejemplo de aplicar la estrategia se puede observar en la Figura 10.

#### 4.1.2. Con *backtracking*

El resultado de aplicar la estrategia descrita anteriormente en el escenario de la Figura 9 se muestra en la Figura 10 y en la Tabla 2.



**Figura 10.** Ejemplo de aplicar la estrategia de exploración al ambiente de la Figura 9.

**Tabla 2.** Información almacenada en el árbol de la figura 10.

Nodo	Obstáculo explorado	Robots	Colas
$n_1$	$O_1$	$r_1, \dots, r_6$	$D_{1,\dots,6} = \{O_1^e, O_6, O_4, O_5, O_2\}$
$n_2$	$O_6$	$r_1, r_5$	$D_{1,5} = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3, O_7\}$
$n_3$	$O_4$	$r_2, r_6$	$D_{2,6} = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3, O_7\}$
$n_4$	$O_5$	$r_3$	$D_3 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3, O_7\}$
$n_5$	$O_2$	$r_4$	$D_4 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3, O_7\}$
$n_6$	$O_3$	$r_1$	$D_1 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_7$	$O_7$	$r_5$	$D_5 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_8$	$O_3$	$r_2$	$D_2 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_9$	$O_7$	$r_6$	$D_6 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_{10}$	$O_3$	$r_3$	$D_3 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_{11}$	$O_3$	$r_4$	$D_4 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_{12}$	$O_7$	$r_3$	$D_3 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$
$n_{13}$	$O_7$	$r_4$	$D_4 = \{O_1^e, O_6^e O_4^e O_5^e, O_2^e, O_3^e, O_7^e\}$

La tabla claramente indica que existen obstáculos que se exploran varias veces por distintos robots. En el peor de los casos todos los robots visitarán todos los obstáculos lo que representa un inconveniente.

Este comportamiento es generado porque los robots una vez que dejan el obstáculo para dirigirse a otro, actualizan su información en función del nuevo obstáculo en el que se encuentren, por lo que después de salir del obstáculo tendrán distinta información y además es difícil que se reúnan nuevamente en algún punto en el ambiente para compartir su progreso.

Para evitar esta situación se pretende forzar a que los robots regresen a cierto obstáculo para compartir la información con robots que hayan regresado a ese mismo obstáculo. Con la información que se comparte, se establecen nuevos obstáculos por explorar y con base a eso se generan nuevos equipos. Lo anterior es posible porque se aprovecha la representación de

la estrategia basada en un árbol. La idea es establecer un nivel que indique el regreso a un obstáculo que en el árbol corresponde a un nodo cuya altura sea múltiplo del valor establecido. Por ejemplo, en el árbol de la Figura 10, supongamos que se establece que los robots regresarán cuando alcancen niveles múltiplos de 2. Como se puede observar una vez que los robots alcancen los nodos  $n_6, n_7, n_8, n_9, n_{10}$  y  $n_{11}$ , están forzados a regresar al nodo  $n_1$  y compartir la información. Como consecuencia  $r_3$  y  $r_4$  encuentran que  $O_7$  ha sido explorado por lo que la exploración termina. A diferencia de la primera estrategia, la exploración da como resultado un árbol de dos niveles.

El Algoritmo 3 describe la nueva estrategia. En este caso primero se aplica la estrategia que describe el algoritmo con la restricción de que acabará mientras haya alcanzado el nivel establecido o todos los obstáculos sean visitados, se mezclan los  $D_i$  y se genera una nueva cola  $D_g$  que mantiene la información de los obstáculos explorados y los que no, hasta ese nivel. Con esa información se generan nuevos equipos y se aplica la estrategia de exploración de obstáculos y se ejecuta recursivamente el Algoritmo 3.

---

**Algoritmo 3** Exploración\_obstáculo\_distinguible\_con\_backtracking

---

**Entrada:** Un ambiente desconocido  $E$ , un equipo  $R$  de  $m$  robots, un obstáculo  $O_s$  que se explorará, el árbol  $T$  de la estrategia, nodo actual  $n_p$  en  $T$  y un *nivel*  
Exploración\_obstáculo\_distinguible( $E, R_i, O_t, T, n_p, nivel$ )  
Regresar el equipo  $R$  al nodo  $n_p$   
Mezclar los  $D_i$  de  $R$ , generando  $D_g$   
Encontrar los Obstáculos que faltan por visitar  $|D_g^{ne}|$   
**si**  $|D_g^{ne}| \neq 0$  **entonces**  
  **si**  $|D_g^{ne}| < n$  **entonces**  
    Crear una cola  $Q$  con los  $D_g$   
    Crear  $R'_i$  tal que  $i = 1, \dots, |D_g^{ne}|$   
    Distribuir los  $m$  robots en los equipos.  
    **para**  $i = 0$  to  $|D_g^{ne}|$  **hacer**  
      Sacar un obstáculo  $O_t$  de  $Q$   
      Mover el equipo  $R'_i$  a  $\partial O_t$   
      Exploración\_obstáculo\_distinguible\_con\_backtracking( $E, R'_i, O_t, T, n_p, nivel$ )  
    **fin para**  
  **si no**  
    Crear la cola  $Q$  con los primeros  $m$  obstáculos no explorados en  $D_g$   
    Crear  $R'_i$  tal que  $i = 1, \dots, m$   
    **para**  $i = 0$  to  $|D_g^{ne}|$  **hacer**  
      Sacar un obstáculo  $O_t$  de  $Q$   
      Mover el equipo  $R'_i$  a  $\partial O_t$   
      Exploración\_obstáculo\_distinguible\_con\_backtracking( $E, R'_i, O_t, T, n_p, nivel$ )  
    **fin para**  
  **fin si**  
**fin si**

---

### 4.1.3. Fusión de las estrategias.

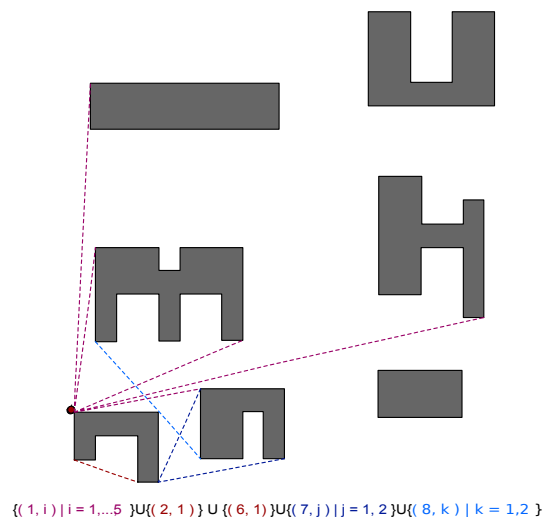
Una tercera estrategia que se propone es combinar las dos anteriores. Primero realizar la exploración hasta alcanzar un nivel arbitrario, los robots regresan para compartir información y finalmente continúan la exploración hasta culminar el proceso. Esto significa que durante toda la exploración sólo se hará un *backtracking*. Se pretende comparar los resultados con las estrategias anteriores.

## 4.2. Algoritmos para obstáculos no distinguibles

En el segundo enfoque abordado en este trabajo, los obstáculos no son distinguibles, esto puede interpretarse como si todos fueran de un mismo color.

Análogamente al enfoque anterior, contamos con  $n$  robots  $r_i$  tal que  $i = 1, \dots, n$ , se explora el obstáculo y se almacenan las bitangentes a medida que se detectan. Inicialmente todos los robots se localizan en el vértice de un obstáculo, como se trata de obstáculos no distinguibles, la única información disponible para los robots son las bitangentes. A medida que exploran los obstáculos se almacena las bitangentes en orden angular de acuerdo a como vayan apareciendo en cada vértice.

Un aspecto a notar es que no es posible distinguir si dos o más bitangentes pertenecen al mismo obstáculo, sin embargo, se distingue el vértice y con eso se diferencian unas de otras en un mismo obstáculo. En la Figura 11 se puede apreciar las bitangentes detectadas en un obstáculo, cada una se almacena de la forma  $(x, y)$  donde  $x$  es el número del vértice y  $y$  la bitangente.



**Figura 11.** Bitangentes en un ambiente con obstáculos no distinguibles.

Los robots tienen limitantes en la comunicación, únicamente lo hacen en *rendezvous*. La dirección en la que se rodean los obstáculos se define antes de iniciar la estrategia. Los robots utilizan las bitangentes para viajar a través de los obstáculos. Cada robot tiene una pila que utiliza para visitar las bitangentes cuando superan el tamaño del equipo. El robot puede guardar y construir su propio árbol.

El robot no conoce su ubicación dentro del ambiente, no tiene acceso a sus coordenadas, por lo que para registrar el progreso de la exploración utiliza marcadores, estos pueden ser de dos tipos:

1. Marcador de inicio: es un marcador único que posee el robot para identificar la posición en la que inicia la exploración de un obstáculo. Este marcador contiene la información del id del robot y está disponible para otros robots.
2. Marcador genérico: es un marcador ilimitado que el robot utiliza para etiquetar los obstáculos que ha visitado. Este marcador contiene la información del robot que visitó el obstáculo.

La estrategia se representa por medio de un árbol, cada nodo contiene la lista de las bitangentes que son visibles desde los vértices de los obstáculos. Las bitangentes representan las aristas del árbol, son el medio para llegar a los obstáculos y sirven para que el robot pueda regresar a los anteriores.

Para entender el algoritmo de exploración (ver Algoritmo 6) se describen las funciones que lo componen, como son la caracterización de los obstáculos, la división de los equipos y la asignación de los obstáculos a explorar.

La exploración de los obstáculos se realiza de la siguiente manera: se selecciona el robot con id menor de todo el equipo, coloca su marcador inicial y rodea el obstáculo, almacena posibles obstáculos por visitar, por medio de las bitangentes que detecta en cada vértice denotado como  $N_c$ .

Este proceso se nombra caracterización del obstáculo (ver Algoritmo 4) y uno de los siguientes casos ocurre durante este proceso:

1. Si el robot no encuentra algún marcador en el obstáculo o únicamente encuentra marcadores iniciales con un id mayor al suyo, entonces  $N_c$  contiene las bitangentes hacia los

posibles obstáculos por explorar. Cuando el robot rodee el obstáculo recoge su marcador inicial y coloca uno genérico indicando que todas las bitangentes asociados al obstáculo han sido detectadas. La información de  $N_c$  se comparte con otros robots en el vértice inicial provocando la actualización del árbol en cada robot.

2. Si el robot encuentra un marcador genérico o un marcador inicial con un id menor al suyo entonces el robot regresa al vértice inicial y recoge su marcador inicial. Si el robot es el único miembro del equipo y en su registro faltan obstáculos por explorar, sigue visitándolos, de otra forma, el robot se apaga o regresa al obstáculo inicial.

---

#### **Algoritmo 4** Caracterización\_del\_obstáculo

---

**Entrada:** Un ambiente desconocido  $E$ , un robot  $r_s$ , una bitangente de llegada  $g_p$ , *nodo\_padre* que es el nodo del último obstáculo visitado.

Sea  $N_c$  el nodo del obstáculo actual.

Sea  $\partial O_s$  el borde del obstáculo actual.

**si**  $g \neq NIL$  **entonces**

El robot marca la bitangente  $g_p$  que conduce al obstáculo anterior visitado.

**fin si**

El robot  $r_s$  coloca su marcador inicial  $p_s$  en el vértice en el que se encuentra.

**repetir**

$r_s$  sigue  $\partial O_s$  y añade un elemento a  $N_c$  por cada bitangente detectada.

**hasta que** Encuentre un marcador genérico, un marcador inicial con un id menor al de  $r_s$  o  $p_s$

**si**  $p_s$  fue encontrado **entonces**

$r_s$  recoge su marcador inicial y coloca un marcador genérico.

**si no**

$N_c \leftarrow NIL$

$r_s$  se dirige a la bitangente de partida y toma su marcador inicial.

**fin si**

**regresar**  $N_c$

---

Una vez caracterizado el obstáculo, se sabe cuales son los posibles obstáculos que serán explorados, la idea es distribuirlos entre los miembros del equipo. La regla para realizar la asignación de los elementos del  $N_c$  se basa en lo siguiente:

1. Si el número de elementos de  $N_c$  es menor a la cantidad de miembros del equipo, este se distribuye por cada elemento de  $N_c$  tal que  $n = |N_c|k + l$  donde  $k > 0$  y  $0 \leq l < n$ . El número de nuevos equipos que se forman está dado por la cardinalidad de  $N_c$ . A cada equipo nuevo se le asigna un elemento de  $N_c$ .
2. En el caso contrario, los elementos de  $N_c$  se dividen entre los miembros del equipo, por lo que el número de los nuevos equipos que se forman esta en función de  $n$ . A cada robot se

le asigna un elemento de  $N_c$ .

El Algoritmo 5 describe la creación de los equipos y la asignación de los obstáculos a explorar. Como se observa, cada equipo repetirá recursivamente la exploración, Algoritmo 6.

---

**Algoritmo 5** Navegación\_del\_obstáculo

---

**Entrada:** Un ambiente desconocido  $E$ , un equipo  $R$  con  $n$  robots y una lista  $N_c$  de obstáculos por visitar.

Sea  $|N_c|$  el número de obstáculos por visitar.

**si**  $|N_c| < n$  **entonces**

Sea  $k, l \in \mathbb{Z}^+$  tal que  $n = |N_c|k + l$  donde  $k > 0$  y  $0 \leq l < n$

Los nuevos equipos son  $\{R_i\}_{i=1}^{|N_c|}$  donde  $k \leq |R_i| \leq k + 1$  y son una partición de  $R$

**para**  $i = 0$  to  $|N_c|$  **hacer**

El  $i$ -ésimo elemento de  $|N_c|$  es asignado al equipo  $R_i$

Sea  $g$  es la bitangente para llegar al  $i$ -ésimo obstáculo en  $N_c$

El equipo  $R_i$  viaja a través de la bitangente  $g$

Exploración\_obstáculos\_indistinguible( $E, R_i, g, N_c$ )

**fin para**

**si no**

Sea  $k, l \in \mathbb{Z}^+$  tal que  $|N_c| = nk + l$  donde  $k > 0$  y  $0 \leq l < |N_c|$

Los nuevos equipos son  $R, \{N_i\}_{i=1}^n$  donde  $k \leq |N_i| \leq k + 1$  y son una partición de  $N_c$

**para todo**  $r_i \in R$  **hacer**

$N_i$  se agrega a la pila de  $r_i$

$r_i$  saca un elemento de su pila y recorre la bitangente  $g$  hasta alcanzar el nuevo obstáculo.

Exploración\_obstáculos\_indistinguible( $E, \{r_i\}, g, N_c$ )

**fin para**

**fin si**

---

El algoritmo de exploración (ver Algoritmo 6), primeramente selecciona el robot con id más pequeño del equipo, al inicio el robot se encuentra en un vértice del obstáculo, por lo que determina si este se encuentra marcado, si contiene una marca, agrega información referente al nodo anterior, la bitangente para llegar al obstáculo y la lista de los posibles obstáculos que faltan por explorar. Si el equipo es solamente un robot este verifica si en la pila donde lleva el registro de los obstáculos que no ha visitado esta vacía, si lo está, no tiene nada mas que hacer y se apaga. De otro modo tiene que sacar un elemento de la pila y realizar la exploración sobre ese obstáculo.

---

**Algoritmo 6** Exploración\_obstáculos\_indistinguible
 

---

**Entrada:** Un ambiente desconocido  $E$ , un equipo  $R$  con  $n$  robots localizados en un vértice del obstáculo, la bitangente  $g$  y el *nodo\_padre* y una lista  $N_c$  de obstáculos por visitar.

Seleccionar el robot  $r_s$  con el id menor de  $R$

**si** El vértice esta marcado **entonces**

Agregar\_ $N_c$ (*nodo\_padre*,  $g$ ,  $NIL$ )

**si**  $n = 1$  y la pila de  $r_s$  no está vacía **entonces**

$r_s$  saca un elemento de su pila ( $q$ )

$r_s$  viaja al obstáculo que le corresponde a  $q$

$r_s$  rodea el obstáculo

$r_s$  viaja a través de la bitangente de  $q$  hasta alcanzar el siguiente obstáculo.

Sea  $b$  la bitangente que  $r_s$  utilizó para alcanzar un nuevo obstáculo

Exploración\_obstáculos\_indistinguible( $E$ ,  $\{r_i\}$ ,  $b$ , *nodo\_padre* de  $q$ )

**si no**

Apagar todos los robots en  $R$

**fin si**

**fin si**

Sea  $N_c$  el conjunto de bitangentes del obstáculo actual

$N_c \leftarrow$  Caracterización\_del\_obstáculo( $E$ ,  $r_s$ ,  $g$ , *nodo\_padre*)

Agregar\_ $N_c$ (*nodo\_padre*,  $g$ ,  $N_c$ )

**si**  $N_c \neq NIL$  **entonces**

Navegación\_del\_obstáculo( $E$ ,  $R$ ,  $N_c$ )

**fin si**

**si**  $n = 1$  y la pila de  $r_s$  no está vacía **entonces**

$r_s$  saca un elemento de su pila ( $q$ )

$r_s$  viaja al obstáculo que le corresponde a  $q$

$r_s$  rodea el obstáculo

$r_s$  viaja a través de la bitangente de  $q$  hasta alcanzar el siguiente obstáculo.

Sea  $b$  la bitangente que  $r_s$  utilizó para alcanzar un nuevo obstáculo

Exploración\_obstáculos\_indistinguible( $E$ ,  $\{r_i\}$ ,  $b$ , *nodo\_padre* de  $q$ )

**fin si**

---

Por otro lado si el vértice en el que se encuentra el equipo no está marcado significa que no ha sido explorado, por lo tanto, es necesario realizar la caracterización (ver Algoritmo 4) y la información resultante se agrega al nodo, junto con la bitangente de llegada y el nodo padre. La caracterización del obstáculo puede dar una lista de bitangentes o una lista vacía, si está vacía se entiende que el obstáculo ya ha sido explorado y lo que resta es verificar si sólo se trata de un robot para revisar los obstáculos de su pila hasta que esta quede vacía, de otra manera navega hasta alcanzar un nuevo obstáculo (ver Algoritmo 5).

Un punto importante a resaltar es que los equipos una vez formados sólo comparten la información con miembros del mismo equipo. Como no es posible distinguir los obstáculos entre sí, no puede efectuarse una estrategia de *backtracking*.

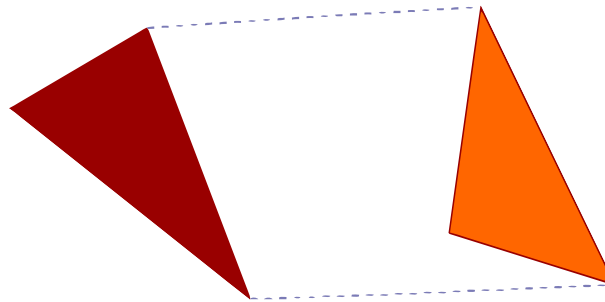


### 4.3. Completitud de las estrategias de exploración

Los algoritmos propuestos garantizan una exploración completa del ambiente, en otras palabras, se rodearán todos los obstáculos en un tiempo finito. En esta sección se muestran resultados para garantizar que las estrategias propuestas terminan en tiempo finito. Además se demuestra que el ambiente es explorado en su totalidad.

**Proposición 4.3.1** *Dados dos triángulos en posición general tales que no se traslapan, existen dos bitangentes entre ellos.*

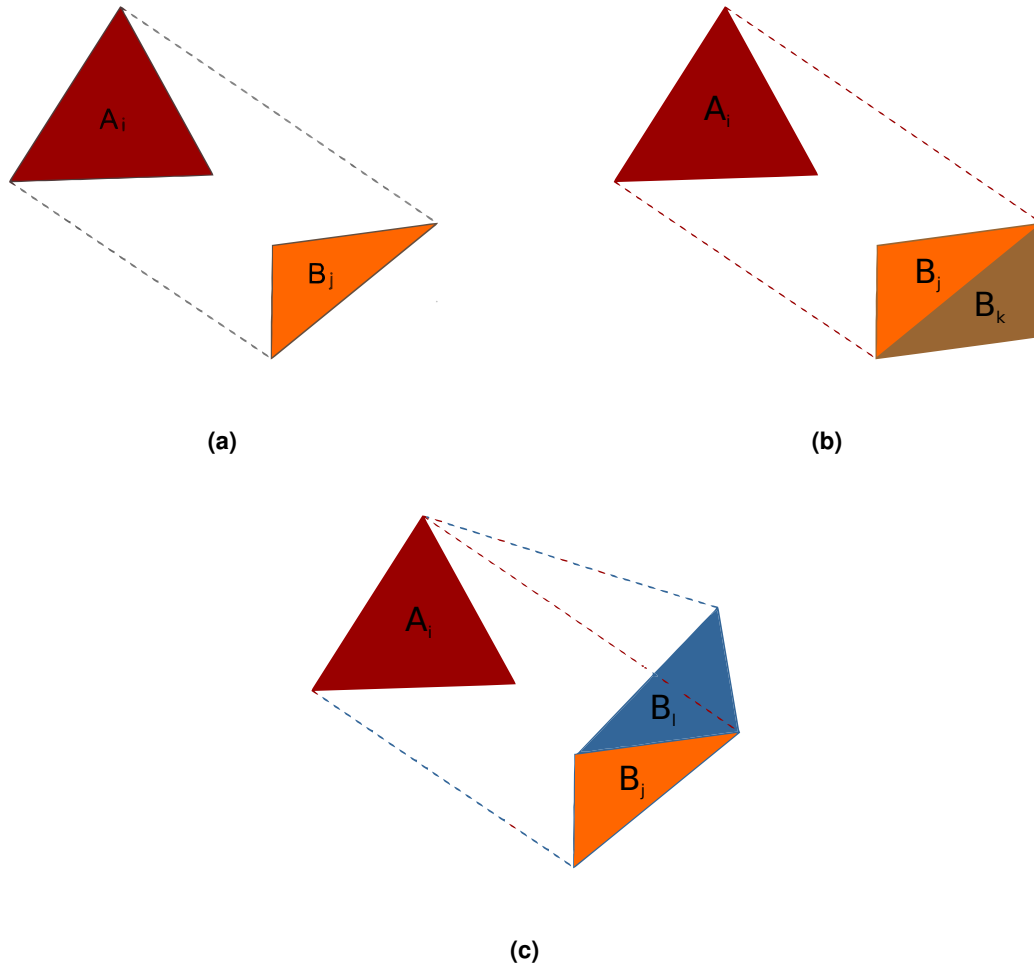
**Demostración.** Ya que los triángulos no se traslapan, podemos construir la envolvente convexa de ambos triángulos. Para encontrarla basta con considerar dos segmentos entre los triángulos. Ambos segmentos son bitangentes generadas por los triángulos. En la Figura 12 se muestran las bitangentes mencionadas. ■



**Figura 12.** Ejemplo de dos de las bitangentes generadas por dos triángulos.

**Proposición 4.3.2** *Dados dos polígonos  $A$  y  $B$  tales que no se traslapan y en posición general, existen al menos dos bitangentes entre ellos.*

**Demostración.** Supóngase que los polígonos  $A$  y  $B$  tienen  $m$  y  $n$  lados respectivamente. Considérese a los conjuntos de triángulos  $\{A_i | i = 1, \dots, m - 2\}$  y  $\{B_i | i = 1, \dots, n - 2\}$  triangulaciones de  $A$  y  $B$ . Es decir  $A = \cup_{i=1}^{m-2} A_i$ , donde  $Area(A_i \cap A_j) = 0 \forall i \neq j$ , y de forma análoga para  $B$ . De manera constructiva, se considerarán dos triángulos, uno de  $A$  y otro de  $B$ . Por la proposición anterior, sabemos que ambos tienen al menos dos bitangentes. Al agregar un triángulo adyacente a alguno de los dos, el número de bitangentes no decrecerá, el nuevo triángulo provoca que desaparezca una existente, pero al mismo tiempo aparece una nueva. En la Figura 13 se muestra gráficamente lo que sucede cuando un triángulo es agregado. Al terminar de formar  $A$  y  $B$ , se conocerán al menos dos bitangentes generadas entre ellos. ■



**Figura 13.** Proceso de agregar nuevos triángulos de la proposición anterior. (b) al agregar el triángulo  $B_k$ , las bitangentes existentes no se ven modificadas. (c) en cambio al agregar  $B_l$  una bitangente desaparece, dando lugar a una nueva.

**Proposición 4.3.3** *Considérese un conjunto de polígonos tales que para cualquier par de ellos existe un camino compuesto de bitangentes y segmentos de la frontera de dichos polígonos. Entonces si otro polígono es agregado, también estará conectado por bitangentes a todos los polígonos originales.*

**Demostración.** Es claro que a pares, cualquier obstáculo del conjunto original tiene dos bitangentes con el polígono agregado. Basta con tomar una bitangente tal que no está obstruida por otros obstáculos. Ya que dicha bitangente conecta al nuevo obstáculo con algunos de los del conjunto original, por lo tanto, el polígono agregado estará conectado con cualquier otro mediante bitangentes y segmentos en las fronteras de los obstáculos. ■

**Teorema 4.3.1** *La frontera de todos los obstáculos es visitada por al menos un robot y además el*

*tiempo de exploración es finito.*

**Demostración.** Por la proposición anterior sabemos que para cualesquiera dos obstáculos existe un camino constituido por frontera y bitangentes generadas. Ya que ambos algoritmos exploran todos los caminos generados por las bitangentes, desde ambos enfoques se visitan todos los obstáculos. Y dado que el número de obstáculos y de bitangentes es finito, entonces la distancia total es finita, por lo que cada equipo de robots termina en tiempo finito la exploración. ■

Con el resultado anterior, podemos demostrar lo siguiente:

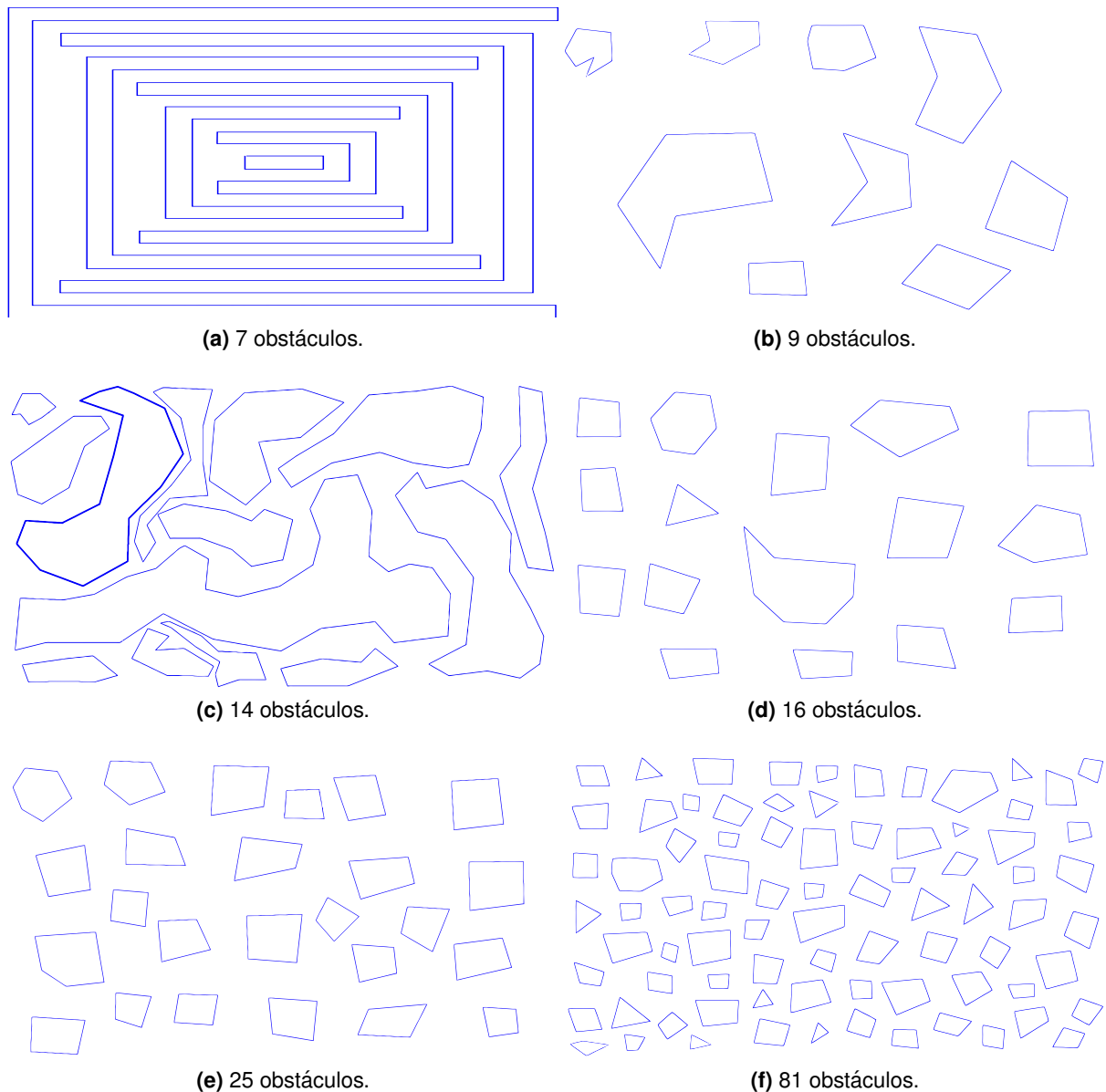
**Teorema 4.3.2** *Para ambos enfoques en los algoritmos de exploración propuestos, todo punto en el espacio libre es visto por al menos un robot.*

**Demostración.** Sea  $x$  un punto del espacio libre, es decir,  $x$  no está en el interior de algunos de los obstáculos. Entonces podemos considerar a  $x$  como un polígono de área cero, tal que  $x$  es su único vértice. Entonces, por la Proposición 4.3.3, sabemos que podemos conectar  $x$  por medio de bitangentes a los obstáculos del ambiente. Entonces existe un camino recto y libre de colisiones, por bitangentes, de los obstáculos hasta  $x$ . Como esto se cumple  $\forall x$  en el espacio libre, es evidente que todo punto del espacio libre es visto por los robots que exploran el ambiente. ■

Con estos resultados, hemos demostrado que todo el ambiente es visto en ambos enfoques: obstáculos distinguibles y no distinguible. Además de que es explorado en tiempo finito.

## Capítulo 5. Experimentos y resultados

En este capítulo, presentamos los experimentos y los resultados obtenidos al implementar y ejecutar los algoritmos propuestos. Los ambientes están compuestos por obstáculos que varían en forma y tamaño. Para los experimentos se utilizan 6 ambientes, que se observan en la Figura 14 y que contienen 7, 9, 14, 16, 25 y 81 obstáculos.



**Figura 14.** Ambientes de prueba.

Cada ambiente difiere en la cantidad de obstáculos, en el perímetro total de estos y en el número de bitangentes que se generan. En la Tabla 3 se muestra la información de los ambientes

**Tabla 3.** Información de los ambientes

Obstáculos	Perímetro	Bitangentes	
		Distinguibles	No distinguibles
7	57200	72	164
9	5687.77	172	258
14	20478.7	94	272
16	7670.53	398	536
25	9762.48	742	950
81	18729.4	2692	3326

en función del número de obstáculos. Se recalca que dependiendo del enfoque se genera un número distinto de bitangentes.

### 5.1. Diseño de los experimentos

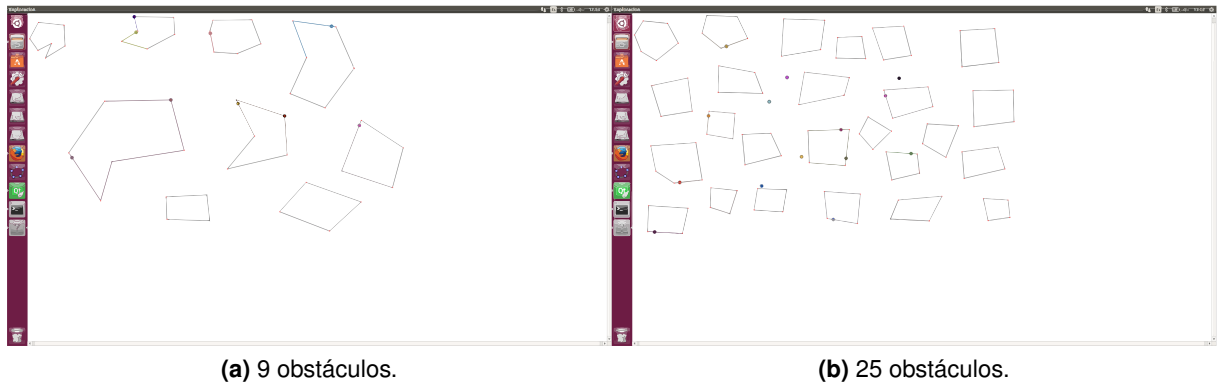
Los algoritmos se implementan en c++ en el *IDE* Qt 5, se utiliza la librería de geometría computacional *CGAL* (The CGAL Project, 2017). El sistema operativo es Ubuntu Linux 14.04.5 LTS. Las pruebas se realizarán en un servidor con las siguientes características:

- Procesador Intel(R) Xeon(R) CPU E7-4809 v3 @ 2.00GHz.
- Memoria RAM 1.5 TB.
- 64 núcleos

Los experimentos se basan en la exploración de distintos ambientes, utilizando los algoritmos propuestos, con el fin de evaluar y comparar el rendimiento tomando como métrica la distancia promedio recorrida por los robots y el promedio de visitas a los obstáculos. Para la exploración de cada ambiente se varía el número de robots en 1, 9, 25 y 144. Estos valores fueron elegidos arbitrariamente con el fin de analizar el comportamiento de las estrategias a medida que aumenta la cantidad de robots.

### 5.2. Ejecución de los algoritmos

Suponiendo que los obstáculos son distinguibles, se realizaron 18 ejecuciones del programa con la estrategia sin *backtracking*. En la Figura 15 se muestran dos ejemplos de la exploración en ambientes con 9 y 25 obstáculos.



**Figura 15.** Ejemplos de la ejecución de la exploración.

Para la estrategia basada en *backtracking* multinivel, se realizaron 144 ejecuciones, distribuidas de la siguiente manera: para cada ambiente se realizaron 24 ejecuciones, 8 por cada conjunto de robots. En las 8, se fue variando el nivel del *backtracking*. Los niveles que se probaron fueron 1, 2, 3, 4, 5, 10, 20 y 30.

Finalmente, para la estrategia con un solo regreso, se tomó en cuenta los resultados de realizar la estrategia sin *backtracking*, por cada prueba se contabilizó la altura del árbol resultante para luego obtener el promedio de las alturas. Para el nivel de regreso se estableció la mitad de ese promedio de las alturas, con la finalidad de que en ningún caso, el árbol quede pequeño y se tenga que realizar al menos un regreso. Las alturas del árbol de la estrategia sin *backtracking* y los promedios se pueden observar en la Tabla 4.

**Tabla 4.** Alturas del árbol y promedios de realizar la exploración sin *backtracking*.

Obstáculos	Robots			Promedio
	9	25	144	
7	4	4	4	4
9	3	3	3	3
14	8	7	6	7
16	7	6	4	5.6
25	16	13	24	17.6
81	72	69	65	68.6

Para el caso donde los obstáculos no son distinguibles se realizaron 24 ejecuciones, por cada ambiente se realizaron 4 pruebas distintas variando el número de robots.

### 5.3. Resultados

Como base para una posterior comparación de las ventajas que nos brinda el realizar la exploración con varios robots, en la Tabla 5 se muestra la distancia que un solo robot recorre para realizar la exploración de cada ambiente.

**Tabla 5.** Exploración de los ambientes con un robot.

Obstáculos	Distancia recorrida
7	122417
9	10629.7
14	25234.3
16	15131.3
25	21491.6
81	44274.7

En las figuras 16, 18, 20 se muestran los promedios de las distancias de la exploración utilizando 9, 25 y 144 robots respectivamente. La línea azul en la parte superior de cada una, muestra la distancia que se recorre cuando se utiliza solamente un robot. Análogamente en las figuras 17, 19 y 21 se muestran los promedios de las visitas a los obstáculos, con el objetivo de analizar si existe alguna relación entre las visitas a los obstáculos y la distancia recorrida. Cada barra de las gráficas muestran los resultados obtenidos al aplicar las distintas estrategias propuestas con sus respectivas variantes.

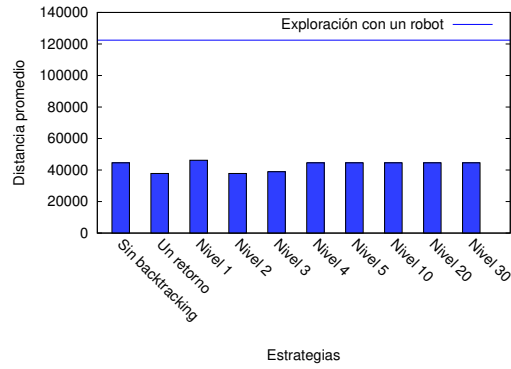
Al explorar el ambiente con 7 obstáculos notamos que independientemente del número de robot se obtiene un comportamiento similar, obteniendo distancias promedio parecidas, de igual forma, el número promedio de visitas es similar. Es importante recalcar que el ambiente es distinto a los demás, ya que contiene obstáculos de mayor perímetro, por lo tanto, la distancia se ve influenciada mayormente por las visitas que por los regresos. Este comportamiento parece ser la razón de estos resultados (ver figuras 16 (a), 18 (a), 20 (a)).

A partir de los resultados de la exploración del ambiente con 9 obstáculos, podemos observar que al hacer la exploración con diferente número de robots, el recorrido de cada uno es menor que al realizar la exploración solamente con uno. También se observa un comportamiento similar en el resto de las estrategias. Centramos nuestra atención en los siguientes cuatro casos de prueba no se realiza *backtracking*, se realiza en niveles 1, 2 o con un retorno solamente. Para 9 robots, podemos notar que realizar *backtracking* es mejor que no realizarlo. Al observar la gráfica de las distancias (ver Figura 16 (b)), notamos que a pesar de que obtenemos resultados muy similares

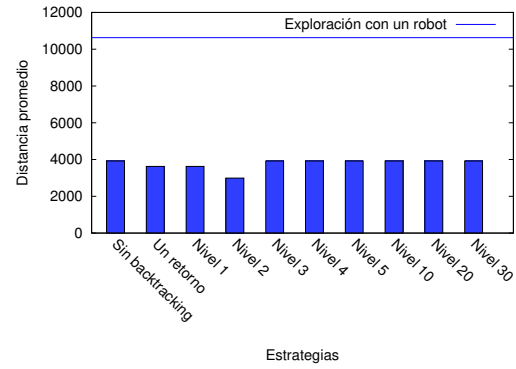
al realizar *backtracking* con nivel 1 y 2, en la gráfica de las visitas (ver Figura 17 (b)) se tiene que estas se reducen con nivel 1, esto se debe al costo en distancia que se necesita para mover los equipos hacia los puntos de reunión. Con 25 robots obtenemos los resultados en las figuras 18 (b) y 19 (b). Podemos observar que en este caso aplicando la estrategia con un retorno y *backtracking* a nivel 1 la distancia recorrida es mayor a la obtenida al no aplicar *backtracking*. Esto es una consecuencia de tener mayor número de robots que de obstáculos resultando contraproducente realizar el movimiento de todos los robots al nivel anterior. Sin embargo, hacerlo con nivel 2 reduce las distancias recorridas pero el número de visitas son mayores que al realizarlo con nivel 1 y con un retorno. Esto tiene sentido ya que al no regresar muy seguido a niveles anteriores, se tendrá mas posibilidades de repetir visitas. Se observó que para 144 robots el comportamiento es similar (ver figuras 20 (b),21 (b)) al obtenido con 25 robots y se piensa que es debido a que el número de robots es mucho mayor que la cantidad de obstáculos.

En el ambiente con 14 obstáculos podemos observar que además de contener obstáculos con un perímetro relativamente grande, para ir de un obstáculo a otro la distancia es considerable. Al ver los resultados de las distancias en las figuras 16 (c), 18 (c) y 20 (c) observamos que varían ligeramente, esto lo relacionamos con el costo de mover los equipo hacia obstáculos anteriores más el perímetro de los obstáculos. Si nos fijamos en las figuras 17 (c), 19 (c) y 21 (c) , veremos que se realiza en promedio la misma cantidad de visitas, sin embargo, las distancias son distintas, por lo que pensamos que se debe al tamaño de los equipos que se mueven a través de los obstáculos.

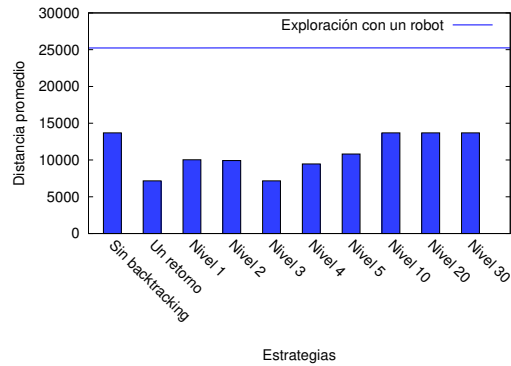




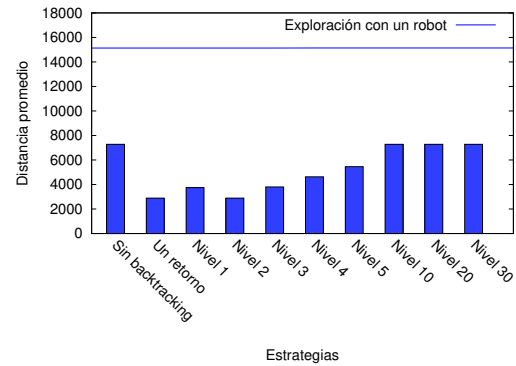
(a) 7 obstáculos



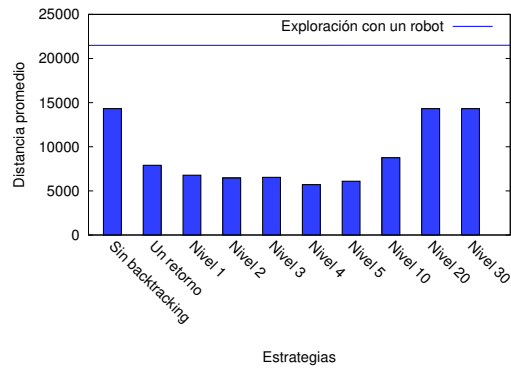
(b) 9 obstáculos



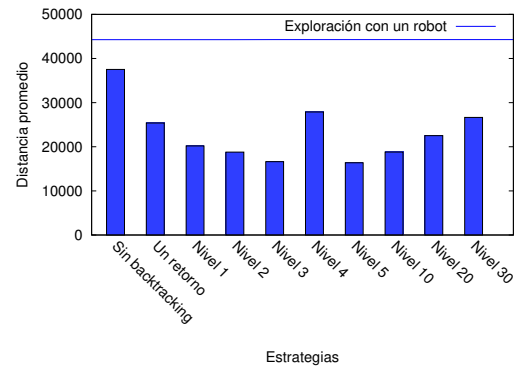
(c) 14 obstáculos



(d) 16 obstáculos

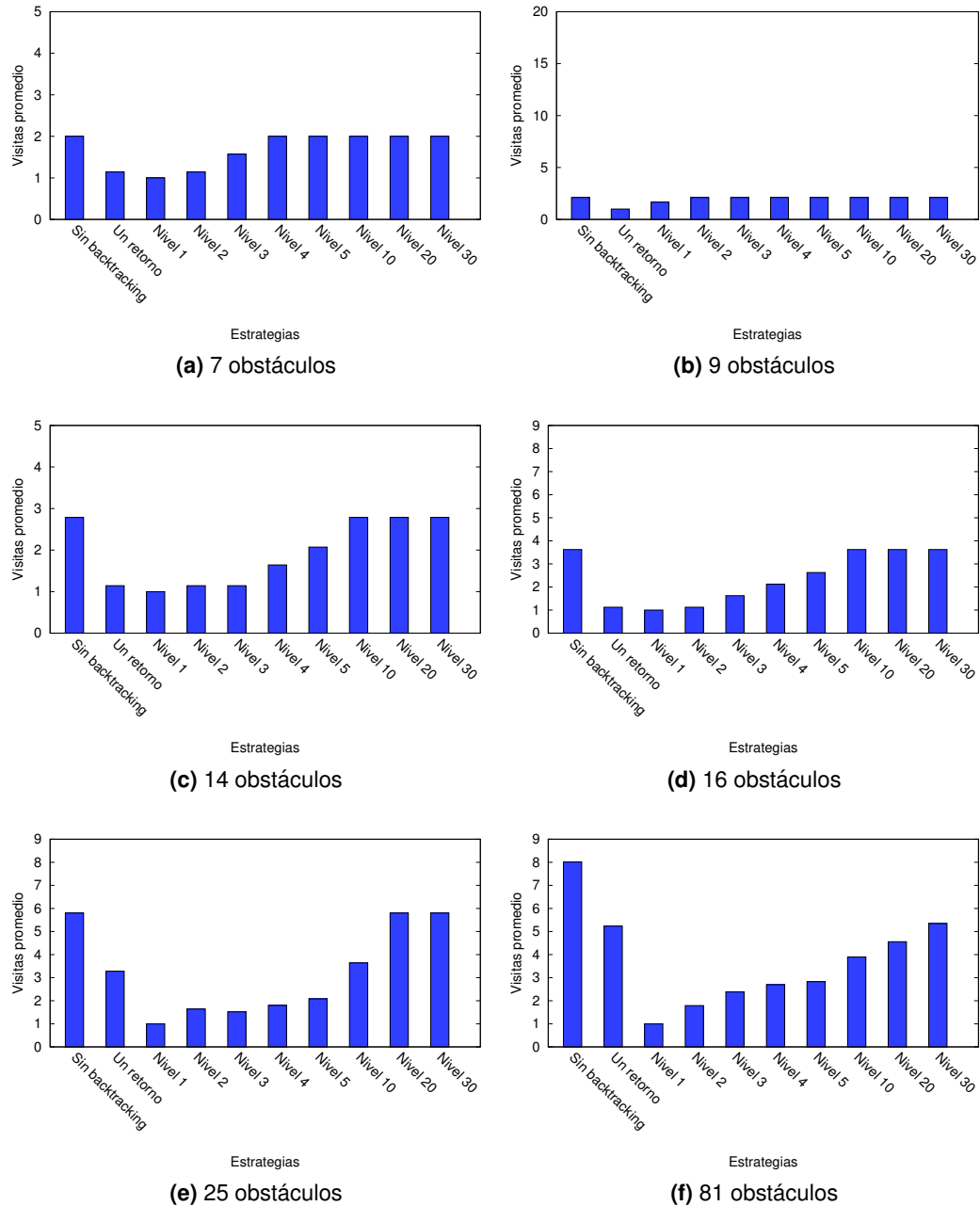


(e) 25 obstáculos



(f) 81 obstáculos

**Figura 16.** Promedio de distancias, utilizando 9 robots.



**Figura 17.** Promedio de visitas, utilizando 9 robots.

Al observar los resultados de las distancias de la exploración del ambiente con 16 obstáculos (ver figuras 16 (d), 18 (d), 20 (d)) nos dimos cuenta que se mejora el rendimiento comparado con la exploración individual del ambiente. Notamos que para este caso, a mayor cantidad de robots las distancias necesarias para la exploración completa del ambiente disminuyen. En general, con 9, 25 y 144 robots los comportamientos resultaron similares. Analicemos los primeros cuatro casos de prueba. En el caso que no se aplica *backtracking*, la distancia es superior al resto de los casos así como también el número de visitas que reporta (por ejemplo, Figura 17 (d)). En los tres casos

siguientes, se observa que la distancia que se recorre con *backtracking* nivel 2, es la menor, pero el número de visitas es mayor, esto esta asociado a que al no regresar frecuentemente al punto de reunión, repite varias visitas, pero se compensa con no regresar al obstáculo anterior. En los casos faltantes se llega a un nivel del *backtracking* en donde la distancia es mayor, y si se opta por un nivel donde el nivel del árbol se quede corto, el resultado se asemeja a no hacer *backtracking*.

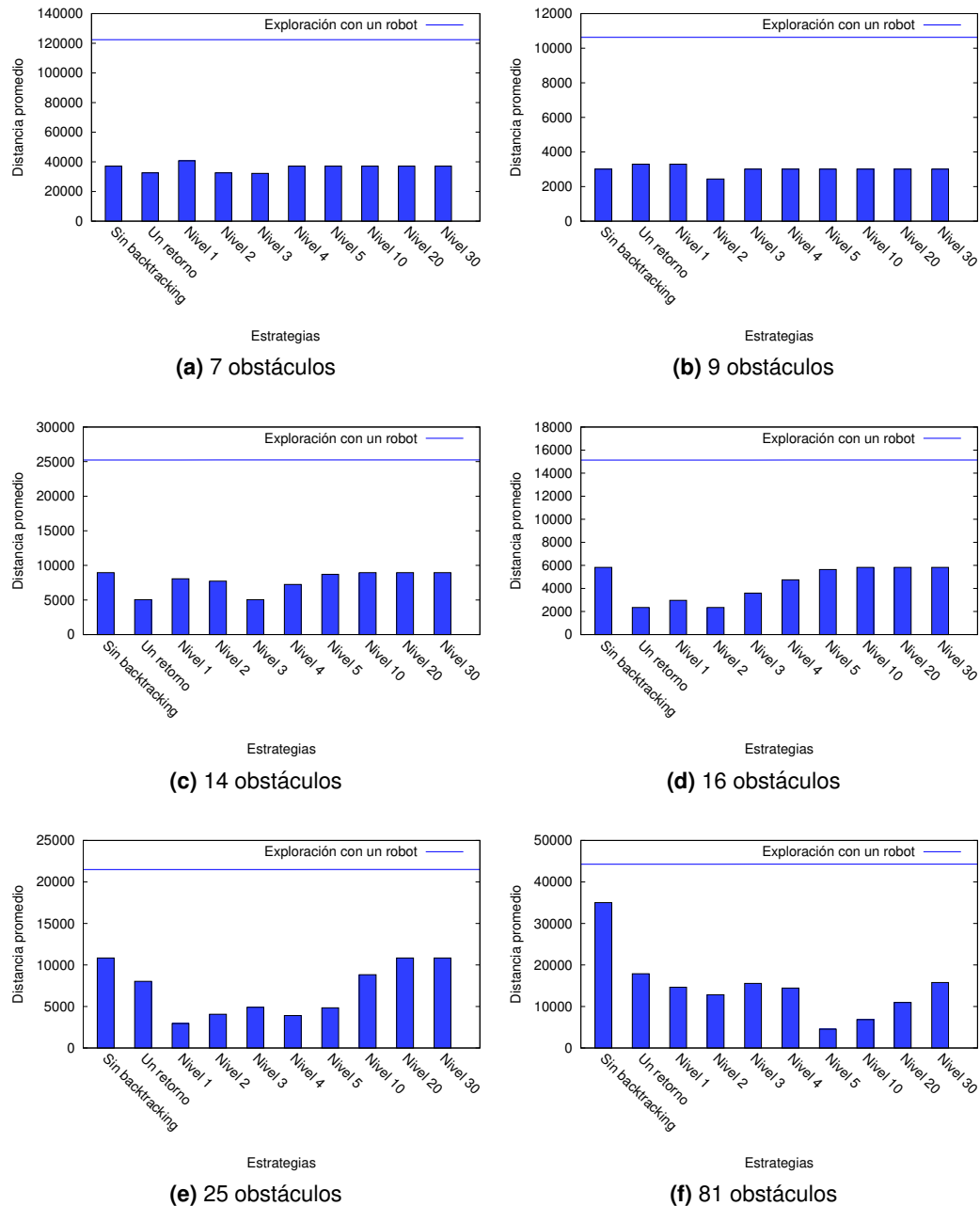
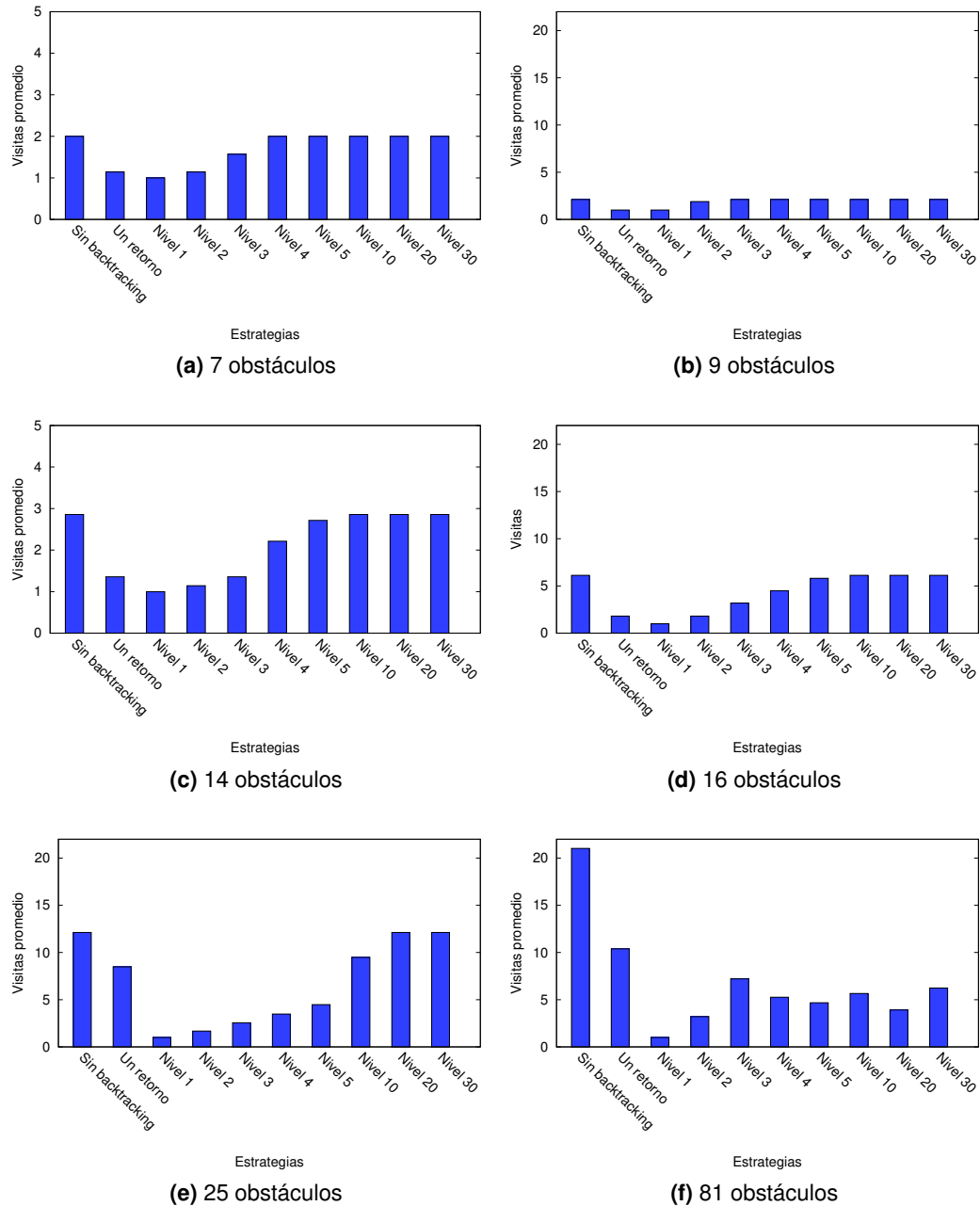


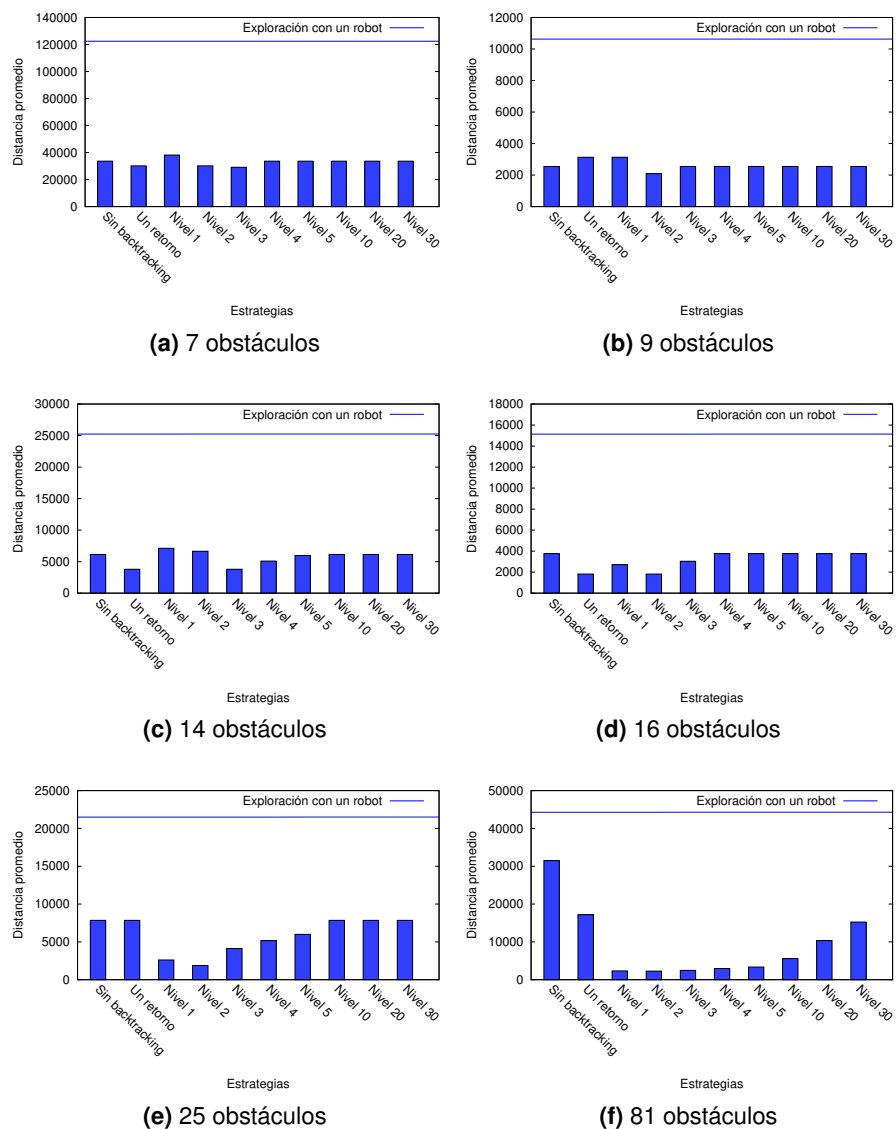
Figura 18. Promedio de distancias, utilizando 25 robots.



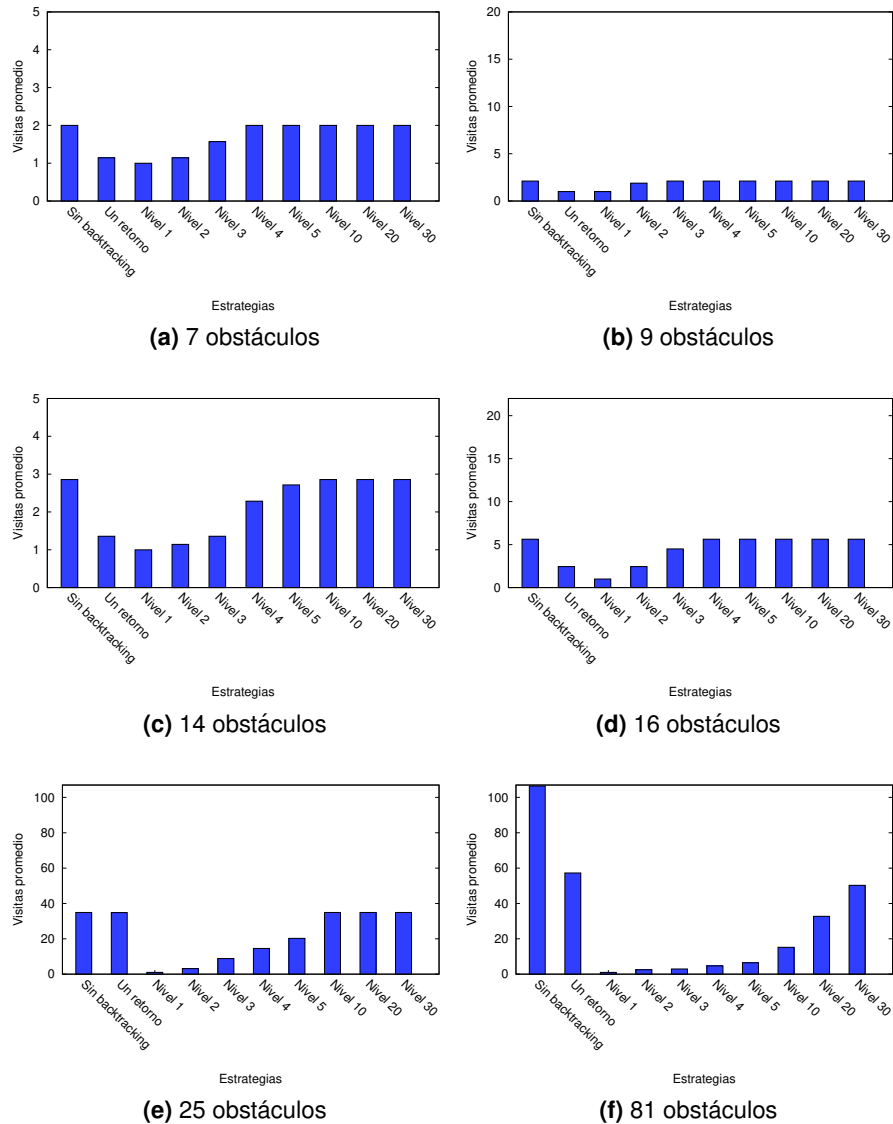
**Figura 19.** Promedio de visitas, utilizando 25 robots.

En las figuras 16 (e), 18 (e) y 20 (e) observamos los resultados de explorar el ambiente con 25 obstáculos utilizando 9, 25 y 144 robots respectivamente. Al igual que en los casos anteriores se mejora el rendimiento individual comparado con la exploración utilizando un robot. Además que la estrategia sin *backtracking* es la que reporta mayores distancias. También nos dimos cuenta que con mayor número de robots se mejora el resultado de las distancias de cada robot. En cuanto a las distintas estrategias, observamos lo siguiente: con 9 robots, utilizando el *backtracking* con nivel 1 y 2, a pesar de obtener resultados casi iguales en distancia, las visitas con nivel 1, son menores

que con nivel 2, las visitas que se ahorran con el nivel 1 se traducen en una mayor distancia por regresar mas veces que con el nivel 2 (ver Figura 17 (e)). Al utilizar 25 robots (ver figuras 18 (e) y 19 (e)), notamos algo distinto, al aplicar la estrategia con nivel 1, la distancia y visitas son menores a las otras estrategias. Pensamos que es debido al número de robots, lo que implica que en cada regreso los robots que se mueven no son muchos, en cambio al utilizar 144 robots la estrategia con nivel 2, arroja el mejor resultado en distancia aunque en visitas es mejor el obtenido con nivel 1 (ver figuras 20 (e) y 21 (e)). Al parecer este comportamiento es causado por el número de robots que tienen que regresarse. Para las demás estrategias a medida que aumentamos el nivel la distancia y las visitas también aumentan.



**Figura 20.** Promedio de distancias, utilizando 144 robots.

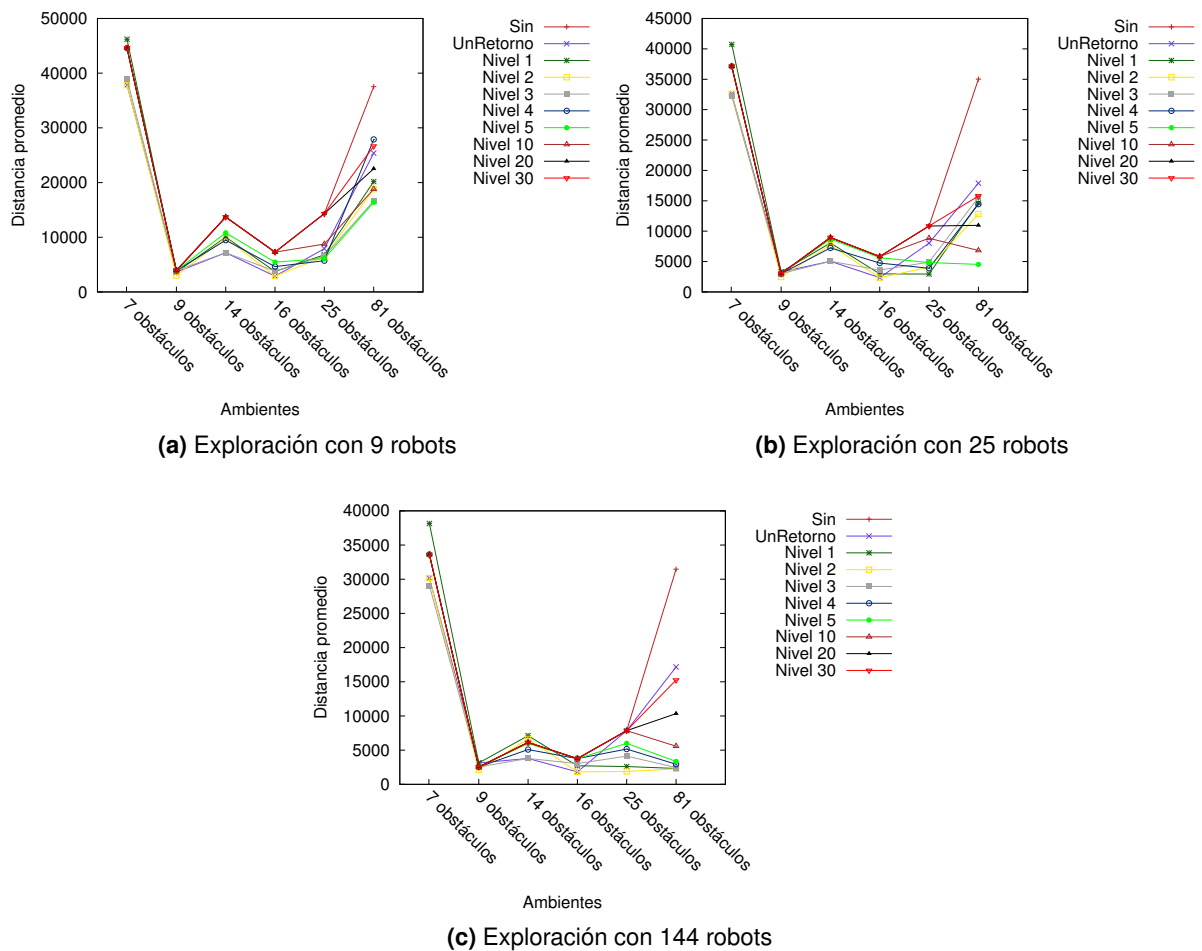


**Figura 21.** Promedio de visitas, utilizando 144 robots.

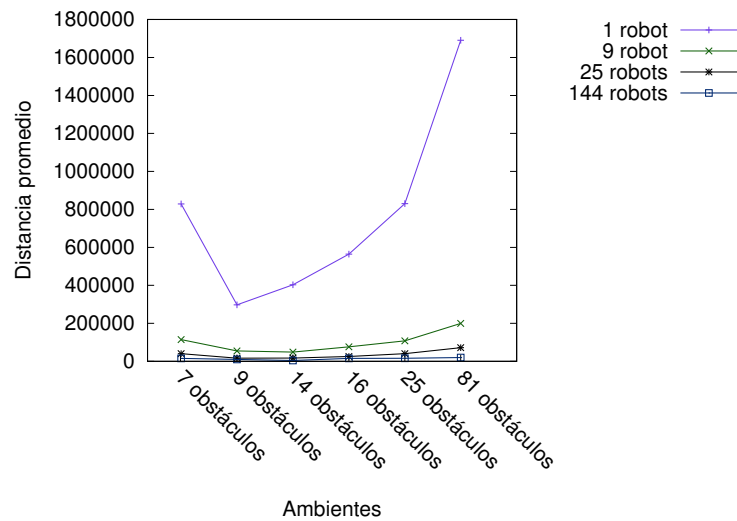
Los resultados de explorar 81 obstáculos se presentan en las figuras 16 (f), 18 (f) y 20 (f). Se observa que cualquier estrategia es mejor que una exploración individual. De nueva cuenta la estrategia sin *backtracking* reporta mayores distancias, también es importante notar que con un número de robots pequeño el desempeño es similar al realizar la exploración con solamente un robot (ver la Figura 16 (f)). Notamos que con *backtracking* nivel 4, la distancia es mayor que utilizar nivel 3, sin embargo, las visitas no varían mucho, esto se atribuye al costo de mover los equipos al punto de reunión. Con 25 robots sucede algo diferente, la estrategia con nivel 5, reporta el mejor resultado en distancia, sin embargo, realiza más visitas (ver figuras 18 (e) y 19 (e)). Se piensa que esto es posible por que a niveles cercanos a la raíz los equipos, es decir el número

de robots es mayor y a medida que avanzan niveles lo más probable es que sus elementos vayan repartiéndose a través de los obstáculos. Observando las gráficas en las figuras 20 (f) y 21 (f), notamos que el mejor resultado al explorar un ambiente con 81 obstáculos con 144 robots, es hacer la exploración con *backtracking* con nivel 2.

Comparando todas las estrategias, podemos observar que si aplicamos la estrategias con *backtracking* con niveles 1, 2 y 3 obtenemos los mejores resultados para la mayoría de las exploraciones. En la Figura 22 observamos que no realizar *backtracking* corresponde a recorrer una distancia mayor. En varios casos realizar *backtracking* a cierto nivel es mejor que hacerlo con 1, 2, o 3 niveles.



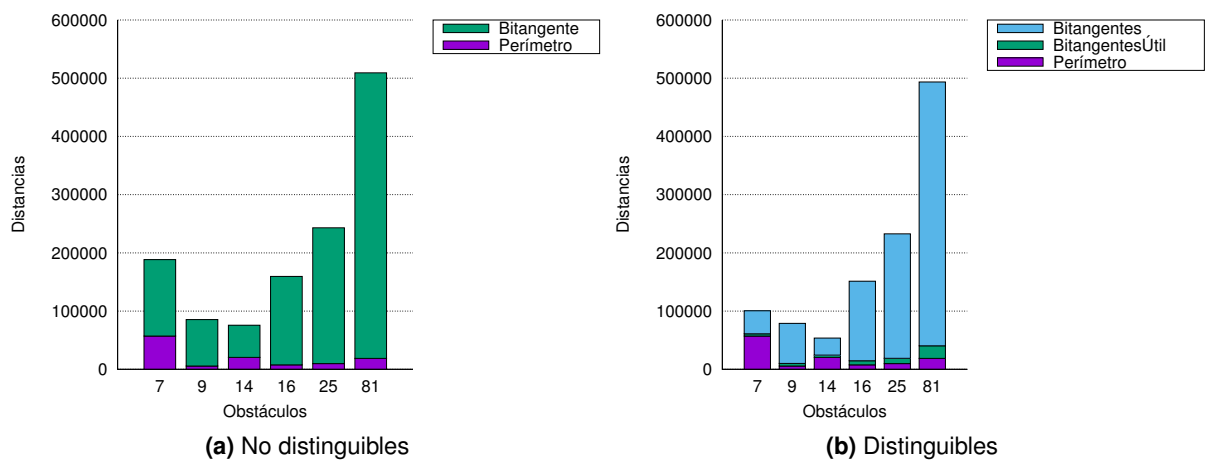
**Figura 22.** Comparativa de las distintas estrategias en obstáculos distinguibles



**Figura 23.** Exploración en ambientes no distinguibles.

Los resultados bajo el enfoque donde los obstáculos no son distinguibles, se presentan en la Figura 23. Se aprecia que con mayor número de robots, la distancia recorrida en promedio disminuye. Sin embargo, es evidente que se recorre mayor distancia que en el enfoque anterior. Esto se debe principalmente a que el número de bitangentes que se genera es mayor que en el enfoque distinguible.

Al comparar los resultados de los enfoques nos dimos cuenta de que al realizar las tareas en obstáculos no distinguibles en la mayoría de los casos reporta peores resultados. En la Figura 24 (a), se puede ver que para el enfoque no distinguible al menos se tiene que recorrer el perímetro de los obstáculos y la distancia acumulada de las bitangentes para culminar la exploración.



**Figura 24.** Distancias correspondientes al perímetro y a las bitangentes

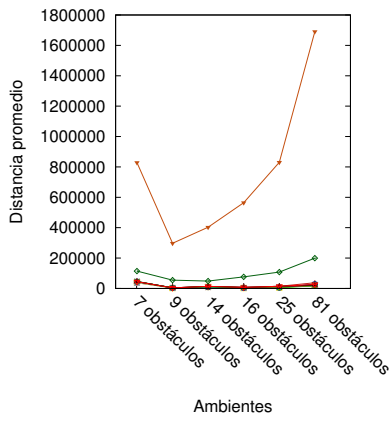


En el enfoque con obstáculos distinguibles no es necesario recorrer todas las bitangentes. En la Tabla 6 se muestra los resultados de comparar la exploración con un sólo robot ya que reportó peores resultados. En la Figura 24 (b), se puede observar que sólo se necesita parte las bitangentes para explorar todo el ambiente. En el enfoque no distinguible se recorre mayor distancia ya que se necesita recorrer todas las bitangentes que se generan dentro del ambiente.

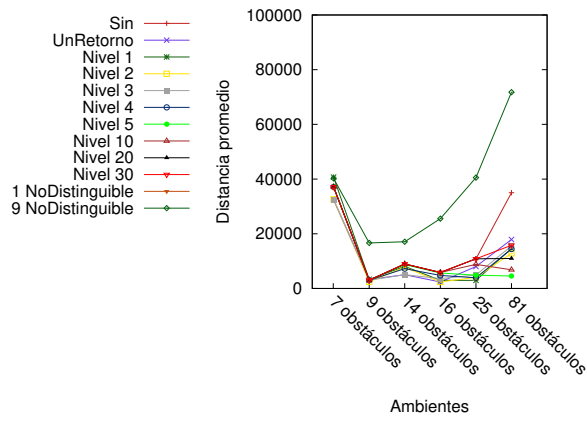
**Tabla 6.** Bitangentes utilizadas con un robot.

Obstáculos	No distinguibles	Distinguibles	Exploración con un robot
7	164	72	6
9	258	172	8
14	272	94	13
16	536	398	15
25	950	742	24
81	3326	2692	80

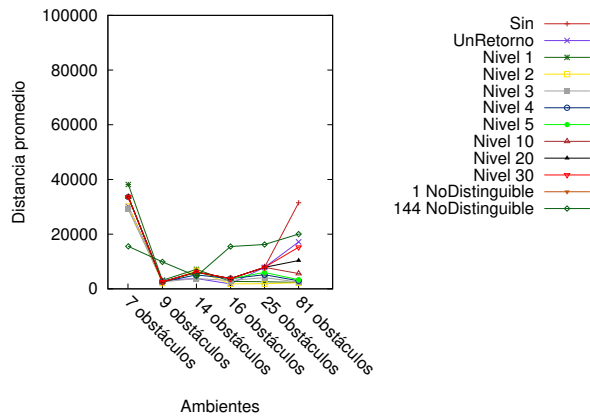
En la Figura 25 se comparan todos los casos de prueba del enfoque distinguible con el que no. Pudimos notar que explorando los ambientes con un robot ofrece peores resultados y al menos se tiene que recorrer todas las bitangentes y el perímetro de todos los obstáculos. La exploración con 9 y 25 robots están lejos del comportamiento del enfoque distinguibles ya que son muchas bitangentes que se necesitan recorrer por los robots. En la Figura 25 (c), se puede observar que con 7 y 14 obstáculos, al parecer obtenemos buenos resultados, esto se debe a que en esos ambientes tenemos menor número de bitangentes que en el resto. En ambos ambientes tenemos pocas bitangentes, sin embargo, los perímetros de los obstáculos son relativamente grandes. En 14 obstáculos sucede algo similar, solo que a diferencia del de 7, la distancia recorrida para terminar la exploración es menor.



(a) Exploración con 9 robots



(b) Exploración con 25 robots



(c) Exploración con 144 robots

Figura 25. Comparativa para los distintos enfoques.

Si tenemos pocas bitangentes en un ambiente no distinguible podemos coseguir buenos resultados.

## Capítulo 6. Conclusiones y trabajo futuro

---

### 6.1. Resumen

En esta tesis, se estudia el problema de exploración de ambientes desconocidos múltiplemente conectados utilizando múltiples robots, con limitantes de comunicación y con sensores que sólo detectan discontinuidades en profundidad. Con estas restricciones se garantiza la culminación de la exploración, además de que con los sensores se utiliza información mínima, se reduce los errores de lectura ya que solamente utiliza las discontinuidades en profundidad, esto ofrece la ventaja de que el robot no desperdicia recursos en procesamiento.

Para solucionar el problema, se proponen estrategias y algoritmos que garantizan la exploración completa del ambiente, se descarta utilizar mapas exactos del ambiente e información acerca de la posición de los robots dentro del ambiente y se opta por construir una estructura de árbol que permita viajar a través de los obstáculos y utilizar únicamente la información de las discontinuidades en profundidad. Se propusieron e implementaron tres estrategias para la exploración de ambientes con obstáculos distinguibles y una para los no distinguibles.

Se diseñaron varios experimentos con el objetivo de comparar las estrategias y determinar cual es la mejor. Primeramente se realizó la exploración de manera individual. Posteriormente, se realizaron experimentos suponiendo que los obstáculos son distinguibles, se tomaron distintos ambientes con diferente número de obstáculos variando el número de robots para las exploraciones. Se observó que en efecto realizar cualquier estrategia es mejor que realizar la exploración de manera individual. Finalmente, se realizaron pruebas para el enfoque en el que los obstáculos no son distinguibles y se observó que se recorre mayor distancia que todas las estrategias con obstáculos distinguibles además que supera al realizarlo con un robot.

### 6.2. Conclusiones

Con base al trabajo de tesis realizado, se concluye lo siguiente:

1. En esta tesis se presenta algoritmos completos para la exploración de ambientes desconocidos múltiplemente conectados con múltiples robots, bajo un enfoque distribuido dado que las tareas de exploración se realizan en equipos y con limitantes en la comunicación

2. Las principales contribuciones de este trabajo son: a) Los algoritmos garantizan la exploración completa del ambiente en un tiempo finito con robots que no tienen información de lo explorado por otros robots. b) Las estrategias toman en cuenta que los robots solamente se pueden comunicar bajo *rendezvous*. En particular en nuestro caso, antes de partir hacia los obstáculos a explorar, estos comparten la información con los elementos del equipo. c) La estrategia se puede escalar a un número finito de obstáculos con diferente número de robots para realizar la exploración.
3. En los experimentos se observó que en todas las ejecuciones, hacer la exploración con un solo robot reporta un recorrido mayor que cualquier estrategia, por lo tanto, la exploración en modo cooperativo aplicando cualquier estrategia, es mejor que explorar con sólo un robot.
4. Sin lugar a duda el *backtracking* disminuye el trabajo redundante durante la exploración, se pudo observar que cuando no se realiza, la distancia y el número de visitas es mayor a las demás estrategias. También se observó que realizar el *backtracking* con niveles cercanos a la raíz, en la mayoría de las pruebas obtuvimos mejores resultados. Otra observación importante es que muchas veces el número de visitas no era factor decisivo para afirmar que la exploración es mejor, debido a que las visitas están en función de la cantidad de robots que regresan para compartir la información. Mover un equipo grande en pocas visitas, aumenta la distancia recorrida.
5. Los experimentos mostraron que las distancias son menores al recorrer ciertas bitangentes que revisar todas. Con obstáculos no distinguibles se necesita visitar todas las bitangentes del ambiente y como se pudo observar las distancias que reportan son mucho mayores que recorrer algunas bitangentes como en el caso de los obstáculos distinguibles.

### 6.3. Trabajo futuro

Algunas ideas para mejorar las estrategias son:

1. En el caso de los obstáculos distinguibles, se pretende proponer y probar otros métodos para que los robots se reúnan y compartan información con el objetivo de reducir las exploraciones redundantes de obstáculos no explorados.
2. En caso de los obstáculos no distinguibles, se pretende investigar estrategias que mejoren y aprovechen la estructura de árbol planteada en este trabajo.

3. Extender las estrategias al caso donde los robots se modelan como discos. Para ello es necesario considerar posibles colisiones entre los robots y oclusiones generadas por los mismos.
4. Se pretende implementar estas estrategias en un robot real.

## Literatura citada

- Asano, T., Ghosh, S. K., y Shermer, T. C. (2000). Chapter 19 – Visibility in the Plane. En: *Handbook of Computational Geometry*. pp. 829–876.
- Buniamin, N., Wan Ngah, W. A. J., y Mohamad, Z. (2014). PointsBug versus TangentBug algorithm, a performance comparison in unknown static environment. En: *2014 IEEE Sensors Applications Symposium (SAS)*, feb. IEEE, pp. 278–282.
- Burgard, W., Moors, M., Stachniss, C., y Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, **21**(3): 376–386.
- Chacón-Rivas, M. y Garita, C. (2013). Mobile course: Development of a mobile app to access university courses information. En: *2013 XXXIX Latin American Computing Conference (CLEI)*, Oct. pp. 1–6.
- Choset, H. M. (2005). *Principles of robot motion : theory, algorithms, and implementation*. MIT Press. p. 603.
- Cortés, J., Martínez, S., Karatas, T., Bullo, F., y Member, S. (2004). Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation*, **20**(2): 243–255.
- de Berg, M., Cheong, O., van Kreveld, M., y Overmars, M. (2008). *Computational Geometry*. Springer Berlin Heidelberg. Berlin, Heidelberg.
- Duan, J., Ren, L., Li, L., y Liu, D. (2016). Moving objects detection in evidential occupancy grids using laser radar. En: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Aug. Vol. 02, pp. 73–76.
- Duckett, T. y Nehmzow, U. (1999). Exploration of unknown environments using a compass, topological map and neural network. *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375)*, pp. 312–317.
- Gage, D. W. (1992). Command Control for Many-Robot Systems. *AUVS-92, the Nineteenth Annual AUVS Technical Symposium*, **10**(June): 15.
- Gąsieniec, L. (2013). Rendezvous Problem. En: *Search Theory*. Springer New York, New York, NY, pp. 167–174.
- Iyengar, S. S. (2007). Feature Extraction and Coverage Problems in Distributed Sensor Networks. Springer, Berlin, Heidelberg, pp. 3–3.
- Kim, S., Bhattacharya, S., Ghrist, R., y Kumar, V. (2013). Topological exploration of unknown and partially known environments. *IEEE International Conference on Intelligent Robots and Systems*, pp. 3851–3858.
- Lavalle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. p. 842.
- Lee, S. K., Fekete, S. P., y McLurkin, J. (2016). Structured triangulation in multi-robot systems: Coverage, patrolling, Voronoi partitions, and geodesic centers. *The International Journal of Robotics Research*, **35**(10): 1234–1260.
- Liang, C.-K. y Chen, Y.-T. (2011). The Target Coverage Problem in Directional Sensor Networks with Rotatable Angles. Springer, Berlin, Heidelberg, pp. 264–273.
- Lopez-Padilla, R., Murrieta-Cid, R., y LaValle, S. M. (2013). Optimal Gap Navigation for a Disc Robot. Springer Berlin Heidelberg, pp. 123–138.

- Márquez, D. F., Torres, I. D., y Luna, J. A. G. (2011). Planificación de movimiento de un robot lego basado en la aplicación del algoritmo Dist-Bug. *Revista Avances en Sistemas e Informática*, **8**(2): 205–210.
- Matignon, L., Jeanpierre, L., y Mouaddib, A. I. (2012). Distributed value functions for multi-robot exploration. En: *2012 IEEE International Conference on Robotics and Automation*, May. pp. 1544–1550.
- Milstein, A. (2008). Occupancy grid maps for localization and mapping. En: *Motion Planning*. InTech.
- Misra, N., Raman, V., Saurabh, S., y Sikdar, S. (2009). The Budgeted Unique Coverage Problem and Color-Coding. Springer, Berlin, Heidelberg, pp. 310–321.
- Ng, J. y Bräunl, T. (2007). Performance Comparison of Bug Navigation Algorithms. *Journal of Intelligent and Robotic Systems*, **50**(1): 73–84.
- O'Rourke, J. (1987). *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc. New York, NY, USA.
- Park, H. y Hutchinson, S. (2016). An Efficient Algorithm for Fault-tolerant Rendezvous of Multi-robot Systems with Controllable Sensing Range.
- Sarmiento, A., Marrieta-Cid, R., y Hutchinson, S. (2004). Planning expected-time optimal paths for searching known environments. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, **1**: 872–878.
- Schwartz, J. T. y Sharir, M. (1983). On the “piano movers” problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, **36**(3): 345–398.
- Taylor, C. J. y Kriegman, D. J. (1995). Vision-based motion planning and exploration. *Algorithmic Foundations of Robotics*, **14**(3): 69–83.
- The CGAL Project (2017). *CGAL User and Reference Manual*. CGAL Editorial Board, edición.
- Tovar, B., Murrieta-cid, R., y Lavelle, S. M. (2007). Distance-Optimal Navigation in an Unknown Environment Without Sensing Distances. *IEEE Transactions on Robotics*, **23**(3): 506–518.
- Veiga, T. S., Miraldo, P., Ventura, R., y Lima, P. U. (2016). Efficient object search for mobile robots in dynamic environments: Semantic map as an input for the decision maker. En: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. pp. 2745–2750.
- Warburton, T. (2017). 3WD10013.
- Wurm, K. M., Stachniss, C., y Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. En: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. pp. 1160–1165.
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. En: *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, Jul. pp. 146–151.
- Yao, Z. y Gupta, K. (2011). Distributed roadmaps for robot navigation in sensor networks. *IEEE Transactions on Robotics*, **27**(5): 997–1004.