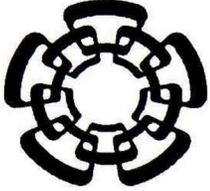


xx (181738.1)



CENTRO DE INVESTIGACIÓN Y
DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO
NACIONAL

COORDINACIÓN GENERAL DE
SERVICIOS BIBLIOGRÁFICOS



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Laboratorio de Tecnologías de Información

**Desplazamiento de objetos por una
comunidad de robots móviles, en
un medio ambiente no estructurado**

Tesis que presenta:

Ezra Federico Parra González

Para obtener el grado de:

**Maestro en Ciencias
en Computación**

Director de la Tesis:
Dr. José Gabriel Ramírez Torres

Cd. Victoria, Tamaulipas, México.

Diciembre, 2008

**CINVESTAV
IPN
ADQUISICION
DE LIBROS**

CLASIF.: I 58.5 .P37 2008
ADQUIS.: 21-207
FECHA: 02/11/2009
PROCED.: Don. 2009

FD. 159505-1001

© Derechos reservados por
Ezra Federico Parra González
2008

Esta investigación fue parcialmente financiada mediante el proyecto No. 51623 del Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas.

This research was partially funded by project number 51623 from "Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas"

La tesis presentada por Ezra Federico Parra González fue aprobada por:

Dr. Adriano de Luca Pennacchia

Dr. José Torres Jiménez

Dr. José Gabriel Ramírez Torres, Director

Cd. Victoria, Tamaulipas, México., 15 de Diciembre de 2008

A mis padres

Agradecimientos

En primer lugar me gustaría agradecer a mi director de tesis, el Dr. José Gabriel Ramírez Torres, cuyo invaluable y generoso apoyo e interés hicieron posible la realización de esta tesis. Debo además expresar mi reconocimiento al Dr. José Torres Jiménez y al Dr. Adriano de Luca Pennacchia por la inestimable y generosa atención en el proceso de revisión de esta tesis.

A mi mamá y a mi papá, a quienes les agradezco su amor, su apoyo incondicional y por enseñarme que la perseverancia y el esfuerzo son el camino para lograr objetivos. Es un privilegio ser su hijo, son los mejores padres. Gracias por estar siempre a mi lado.

Muy en especial quiero agradecer a Von, por su compañía, sus consejos, su comprensión y por motivarme a seguir siempre adelante. Gracias por estar conmigo.

Por supuesto agradezco a mis hermanos por la compañía y el apoyo que me brindan. Sé que cuento con ustedes siempre.

También quiero agradecer a todos los profesores del laboratorio, Dr. José Torres, Dr. Gregorio Toscano, Dr. Iván López, Dr. Víctor Sosa, Dr. Claudio Castellanos, Dr. Eduardo Rodríguez y muy en especial al director del laboratorio, el Dr. Arturo Díaz Pérez por confiar en mí y darme la oportunidad de ser parte de esta institución.

Así mismo, hago extensivos estos agradecimientos a todos mis compañeros y amigos del laboratorio por todo el tiempo y las vivencias que hemos compartido.

Quiero dar las gracias a toda mi familia, a mis amigos y a toda esa gente que sería difícil de mencionar individualmente, pero que de una u otra manera fueron partícipes de éste, nuestro éxito, gracias a todos.

Finalmente agradezco al Cinvestav-Tamaulipas y al Consejo Nacional de Ciencia y Tecnología CONACyT por el apoyo y patrocinio.

Índice General

| | |
|---------------------------------------------------------------------------------------------------|-------------|
| Índice General | I |
| Índice de Figuras | v |
| Índice de Tablas | ix |
| Índice de Algoritmos | xi |
| Publicaciones | xiii |
| Resumen | xv |
| Abstract | xvii |
| 1. Introducción | 1 |
| 1.1. Desplazamiento de objetos por comunidades de robots móviles (“Box-pushing problem”). | 3 |
| 1.2. Planteamiento del problema | 5 |
| 1.3. Objetivos del proyecto | 6 |
| 1.4. Breve descripción del enfoque propuesto | 7 |
| 1.5. Conjunto de hipótesis iniciales | 9 |
| 1.6. Estructura del reporte de tesis | 10 |
| 2. Estado del arte | 11 |
| 2.1. Historia de la robótica móvil | 11 |
| 2.2. Tipo de robots y de movilidad | 17 |
| 2.2.1. Robots móviles a ruedas | 18 |
| 2.2.2. Robots móviles a patas | 18 |
| 2.2.3. Robots móviles híbridos | 19 |
| 2.2.4. Robots móviles voladores | 20 |
| 2.3. Problemas asociados a la robótica móvil | 20 |
| 2.4. Problema del movimiento del piano (piano mover’s problem) | 21 |
| 2.5. Revisión bibliográfica | 24 |
| 2.5.1. Esquemas existentes | 24 |
| 2.5.2. Esquemas basados en el uso de espacios de configuraciones | 28 |
| 2.6. Resumen del capítulo | 30 |

| | |
|-------------------------------------------------------------------------------|-----------|
| 3. Representación del espacio | 31 |
| 3.1. Transformaciones del espacio | 32 |
| 3.2. Intersecciones entre polígonos (colisiones) | 37 |
| 3.2.1. Intersección entre una línea y un polígono | 37 |
| 3.3. Espacio de configuraciones "C-space" | 43 |
| 3.3.1. Expansión virtual de objetos | 44 |
| 3.3.2. Discretización del espacio de configuraciones | 46 |
| 3.4. Resumen del capítulo | 47 |
| 4. Cálculo de trayectorias | 51 |
| 4.1. Problema del cálculo de trayectorias | 52 |
| 4.2. Problema de la planeación de movimientos | 54 |
| 4.2.1. Características de la planeación | 54 |
| 4.3. Algoritmo de frente de onda | 56 |
| 4.3.1. Construcción del frente de onda | 57 |
| 4.4. Modelo de navegación | 60 |
| 4.4.1. Reconfiguraciones | 61 |
| 4.4.2. Selección de vecinos | 62 |
| 4.4.3. Heurística de "Medición del alcance" | 64 |
| 4.4.4. Heurística de "Medición del alcance con validación de capas" | 66 |
| 4.4.5. Heurística de "Construcción de grafo ruta" | 72 |
| 4.5. Obtención de puntos de empuje | 75 |
| 4.6. Algoritmo A* (A-Estrella) para el cálculo de trayectorias | 76 |
| 4.6.1. Heurística A* modificada | 80 |
| 4.7. Resumen del capítulo | 80 |
| 5. Resultados | 83 |
| 5.1. Esquemas a prueba | 84 |
| 5.2. Descripción de los experimentos. | 85 |
| 5.2.1. Ambientes | 87 |
| 5.2.2. Métricas | 90 |
| 5.2.3. Ambiente de implementación | 91 |
| 5.3. Resultados | 91 |
| 5.3.1. Primera fase de pruebas | 91 |
| 5.3.1.1. Reconfiguraciones resolución 50cm | 92 |
| 5.3.1.2. Tiempo resolución 50cm | 93 |
| 5.3.2. Segunda fase de pruebas | 94 |
| 5.3.2.1. Reconfiguraciones resolución 20cm | 95 |
| 5.3.2.2. Tiempo resolución 50cm | 96 |
| 5.3.3. Tercera fase de pruebas | 96 |
| 5.4. Casos específicos. | 97 |

| | |
|------------------------------------------|------------|
| 5.5. Resumen de resultados | 100 |
| 5.6. Resumen del capítulo | 100 |
| 6. Conclusiones y trabajo futuro. | 103 |
| 6.1. Trabajo futuro | 106 |
| A. Resultados | 109 |
| Bibliografía | 127 |

Índice de Figuras

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1. Esquema clásico del problema "box-pushing" | 4 |
| 1.2. Esquema básico de la estrategia propuesta. | 8 |
| 2.1. Mars-Rover. | 14 |
| 2.2. Robot Pioneer P3-DX. | 16 |
| 2.3. Robot Lego NXT. | 17 |
| 2.4. Movimiento de un piano usando robots móviles. | 22 |
| 2.5. "Object closure" | 26 |
| 2.6. "Pusher-Watcher" | 27 |
| 2.7. "swarm intelligence" | 27 |
| 2.8. Ejemplo de trayectoria | 29 |
| 2.9. Ejemplo de trayectoria | 29 |
| 3.1. Descripción vectorial del mundo en su sistema coordenado. | 33 |
| 3.2. Descripción vectorial del objeto en su sistema coordenado. | 33 |
| 3.3. Descripción vectorial del objeto y de los obstáculos en un mismo sistema coordenado. | 34 |
| 3.4. Representación gráfica de los vectores que definen las restricciones. | 37 |
| 3.5. Obtención de un segmento de línea que satisface las restricciones de un polígono. | 38 |
| 3.6. Ejemplo de 3 segmentos de línea que satisfacen el conjunto de restricciones de ambos polígonos. | 41 |
| 3.7. Configuración del objeto. | 44 |
| 3.8. Ejemplo de la expansión virtual de un objeto. | 45 |
| 3.9. Ejemplo de 2 trayectorias idénticas, la Figura 3.9 A es representada con el tamaño original del objeto, 3.9 B es representada con el objeto expandido. | 46 |
| 3.10. A) Representación gráfica del C-space con una caja rotada 10 grados. B) Representación binaria del C-space correspondiente a la Figura A. | 49 |
| 3.11. Cadena de configuraciones válidas que forman una ruta para desplazar un objeto de un punto a otro. | 50 |
| 4.1. Ejemplo de una matriz de marcada con el algoritmo de frente de onda. | 59 |
| 4.2. Ejemplo de una ruta construida a partir del C-space marcado con el algoritmo del frente de onda. | 60 |
| 4.3. Ejemplo de una vecindad de 4 y 6 elementos. | 61 |
| 4.4. Ejemplo de una ruta con múltiples reconfiguraciones | 63 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.5. Rutas calculadas con las heurísticas de selección de vecino aleatorio A y evaluación en orden predefinido B | 64 |
| 4.6. A) Representación gráfica de la trayectoria con la heurística de selección de vecinos predefinido. B) Trayectoria obtenida con la aplicación de la heurística propuesta para reducción de reconfiguraciones. | 66 |
| 4.7. Ejemplo de 2 configuraciones distintas de la caja en donde los puntos de empuje son afectados por la rotación. | 68 |
| 4.8. Ejemplo de rutas generadas con las heurísticas de "Medición de alcance" y "Medición de alcance con validación de capa" | 72 |
| 4.9. Ejemplo de construcción del árbol de ruta. | 73 |
| 4.10. A) Ejemplo de ruta construida con la heurística de "Grafo de ruta". B) Puntos de empuje válidos para la ruta. | 75 |
| 4.11. A) Ruta calculada con el algoritmo A* usando la heurística de reducción de distancia. B) Ruta calculada con el algoritmo A* usando la heurística de reducción de cambios de dirección. | 81 |
| 5.1. Ambiente de pruebas 1, con 9 obstáculos y un 44.4 % de área libre. | 88 |
| 5.2. Ambiente de pruebas 2, con 9 obstáculos y un 49 % de área libre. | 88 |
| 5.3. Ambiente de pruebas 3, con 12 obstáculos y un 56.4 % de área libre. | 89 |
| 5.4. Ambiente de pruebas 4, con 11 obstáculos y un 67.9 % de área libre. | 89 |
| 5.5. Ambiente de pruebas 5, con 12 obstáculos y un 79 % de área libre. | 90 |
| 5.6. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 50cm | 93 |
| 5.7. Gráfica de tiempo por heurística. Ambiente 4. Resolución 50cm. Tiempos expresados en <i>ms</i> | 94 |
| 5.8. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 20cm | 96 |
| 5.9. La ruta A | 98 |
| 5.10. Ejemplo de dos frentes de onda que se encuentran entre ellos. | 99 |
| A.1. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 1. Resolución 20cm. Ver Tablas A.1 y A.2 | 110 |
| A.2. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 2. Resolución 20cm. Ver Tablas A.3 y A.4 | 110 |
| A.3. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 3. Resolución 20cm. Ver Tablas A.5 y A.6 | 111 |
| A.4. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 20cm. Ver Tablas A.7 y A.8 | 111 |
| A.5. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 5. Resolución 20cm. Ver Tablas A.9 y A.10 | 112 |

A.6. Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4.
Resolución 50cm. Ver Tablas A.11, A.12 y A.13 112

Índice de Tablas

| | |
|----------------------------------------------------------------------------------------------------|-----|
| 5.1. Cantidad de pruebas realizada por cada heurística | 85 |
| 5.2. Cantidad de pruebas realizada por cada ambiente | 86 |
| 5.3. División de pruebas ambiente\heurística. Resolución 20cm | 86 |
| 5.4. División de pruebas ambiente\heurística. Resolución 50cm | 87 |
| 5.5. Características de los ambientes de pruebas | 90 |
| 5.6. Reconfiguraciones por heurística. Ambiente 4. Resolución 50cm | 92 |
| 5.7. Tiempo por heurística. Ambiente 4. Resolución 50cm. Tiempos expresados en <i>ms</i> | 94 |
| 5.8. Reconfiguraciones por heurística. Ambiente 4. Resolución 20cm | 95 |
| 5.9. Tiempo por heurística. Ambiente 4. Resolución 20cm. Tiempos expresados en <i>ms</i> | 96 |
| | |
| A.1. Ambiente 1. Resolución: 20cm. Propagación de frente de onda: 4305ms | 113 |
| A.2. Ambiente 1. Resolución: 20cm. Propagación de frente de onda: 4305ms | 114 |
| A.3. Ambiente 2. Resolución: 20cm. Propagación de frente de onda: 2082ms | 115 |
| A.4. Ambiente 2. Resolución: 20cm. Propagación de frente de onda: 2082ms | 116 |
| A.5. Ambiente 3. Resolución: 20cm. Propagación de frente de onda: 831ms . | 117 |
| A.6. Ambiente 3. Resolución: 20cm. Propagación de frente de onda: 831ms . | 118 |
| A.7. Ambiente 4. Resolución: 20cm. Propagación de frente de onda: 2996ms | 119 |
| A.8. Ambiente 4. Resolución: 20cm. Propagación de frente de onda: 2996ms | 120 |
| A.9. Ambiente 5. Resolución: 20cm. Propagación de frente de onda: 2999ms | 121 |
| A.10. Ambiente 5. Resolución: 20cm. Propagación de frente de onda: 2999ms | 122 |
| A.11. Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms . | 123 |
| A.12. Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms . | 124 |
| A.13. Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms . | 125 |

Índice de Algoritmos

| | | |
|----|---------------------------------------------------------------------|----|
| 1. | Transformación lineal | 34 |
| 2. | Calcula inecuaciones | 36 |
| 3. | Calcula colisiones | 42 |
| 4. | Calcula C-space | 48 |
| 5. | Propagación de frente del onda | 59 |
| 6. | Heurística 1 "Medición de alcance" | 67 |
| 7. | Heurística 1 "Medición de alcance con validación de capa" | 71 |
| 8. | Heurística 3 "Grafo de ruta" | 74 |
| 9. | A-Estrella | 79 |

Publicaciones

Ezra Federico Parra-González, Gabriel Ramírez-Torres, Gregorio Toscano-Pulido, *Motion Planning for Cooperative Multi-Robot Box-Pushing Problem*, 11th Ibero-American Conference on Artificial Intelligence, Advances in Artificial Intelligence Iberamia 2008 LNAI 5290, ISBN 3-540-88308-8, Ed. Springer, 2008, pp. 382-391

Ezra Federico Parra-González, Gabriel Ramírez-Torres, *Cooperative Multi-Robot Box-pushing in a Cluttered Environment*, Proceedings IEEE Electronics, Robotics and Automotive Mechanics Conference 2008, Cuernavaca, México, octubre 2008, pp. 514-519.

Ezra Federico Parra-González, Gabriel Ramírez-Torres, Gregorio Toscano-Pulido, *Desplazamiento de Objetos por una Comunidad de Robots Móviles en un Medio Ambiente no Estructurado*, 9no. Simposium Iberoamericano de Computación e Informática SICI 08, octubre 2008, Aguascalientes, México, pp. 349-360.

Resumen

Desplazamiento de objetos por una comunidad de robots móviles, en un medio ambiente no estructurado

por

Ezra Federico Parra González

Maestro en Ciencias del Laboratorio de Tecnologías de Información
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2008
Dr. José Gabriel Ramírez Torres, Director

Con el avance de la tecnología, parece inevitable que en un futuro no muy lejano será algo ordinario encontrar robots en nuestros hogares. Es necesario recordar que, como muchos otros inventos de la humanidad, la robótica nació del deseo del ser humano de tener un ente artificial que realice su trabajo. Por supuesto, resultará indispensable que un robot sea capaz de desplazar objetos de un punto a otro de nuestro hogar.

El problema de desplazar objetos, únicamente empujándolos, por una comunidad de robots móviles (box-pushing) en un medio ambiente no estructurado ha demostrado ser un problema muy complejo pero con múltiples aplicaciones prácticas (e.g. automatización de procesos industriales, rescate de personas, movimiento de objetos peligroso), y que representa un reto muy interesante para el desarrollo de esquemas de coordinación multiagente.

En esta tesis se presenta una nueva estrategia para resolver este problema, inspirada en el algoritmo del frente de onda, pero que incluye ciertas consideraciones originales para obtener trayectorias que facilitan el desplazamiento del objeto con robots móviles no especializados. El método propuesto puede calcular rutas cortas para mover objetos irregulares, reduce los cambios de dirección del objeto, reduce los tiempos de desplazamiento y realiza una búsqueda de puntos de empuje para los robots involucrados.

Abstract

Cooperative Multi-Robot Box-pushing in a Cluttered Environment

by

Ezra Federico Parra González

Master of Science in Laboratory of Information Technology

Research Center for Advance Study from the National Polytechnic Institute, 2008

Dr. José Gabriel Ramírez Torres, Advisor

It seems that technological breakthrough inevitably will lead us in the near future to ordinarily find robots in our homes. It is necessary to remember that like many other human inventions, the robotics was born from the human desire to have an artificial being that carries out their work. Obviously, it will be indispensable that robots be able to move objects from one place to other inside our homes.

The problem to move objects, only pushing them and made by a community of mobile robots (known as box-pushing) in cluttered environments, has demonstrated to be a very complex problem but with multiple practical applications (e.g. industrial processes automatization, people rescue, movement of dangerous objects), and represents a very interesting challenge for the development of coordination schemes for multiagent systems.

In this thesis we present a new strategy to solve this problem, inspired from the wave front algorithm but including certain original considerations which help to obtain trajectories that facilitate the movement of the object by unspecialized robots. The proposed method is able to calculate shortest routes to move irregular objects, minimising the changes of direction of the object, reducing the movement times and determining a search of the best pushing points for involved robots.

1

Introducción

¿Robots en nuestras vidas?

“Los robots siguen la misma trayectoria que recorrieron las computadoras aunque, hablando en términos generales, con un retraso de 20 a 25 años.” “Si se mantiene el paralelismo primitivo entre ordenadores y robots, cabe esperar aplicaciones masivas de robots en los próximos diez a quince años. Hacia 2020 estarán entonces presentes en cualquier aspecto de nuestra existencia.”

Rodney Brooks¹

Actualmente, es ampliamente aceptado que los sistemas multi-robot presentan múltiples ventajas sobre los sistemas basados en un único robot [4]. Las motivaciones más comunes para desarrollar sistemas multi-robot son:

¹Fragmentos tomados del libro *Cuerpos y Máquinas* [2]

1. Un sistema multi-robot puede realizar tareas más complejas que un sistema mono robot.
2. Por lo regular la carga de la ejecución de las tareas puede ser distribuida entre los elementos de la comunidad de robots.
3. El dividir una actividad en múltiples agentes permite construir modelos poderosos sin necesidad de contar con un único agente altamente especializado.
4. Una comunidad de robots puede resolver problemas de manera más rápida, explotando el paralelismo inherente a un sistema distribuido.
5. La introducción de múltiples robots permite tener sistemas más robustos a través de la redundancia.

En esta tesis desarrollamos un esquema original para la resolución del problema del movimiento de objetos por comunidades de robots móviles (box-pushing). Por la amplitud de la solución propuesta, en este trabajo nos concretamos al diseño de una estrategia que permite resolver el problema del cálculo de trayectorias para los objetos que serán desplazados por los robots móviles que, a diferencia de otros trabajos reportados en la literatura, integra el evitamiento de colisiones con los obstáculos del medio ambiente.

La dificultad del problema de box-pushing queda de manifiesto por la necesidad de controlar las múltiples fuerzas que se aplican sobre la caja (fricción, dirección y fuerza de los puntos de contacto con los robots, peso, etc.). Esta tarea es aún más difícil si se toman en cuenta las entidades dinámicas que forman parte del ambiente (robots, obstáculos, objetos a empujar, etc.).

En este capítulo se planteará de manera concisa el problema que se desea resolver, se mostrarán algunas de las aplicaciones prácticas más relevantes y se describirá brevemente

la estrategia que se propone en esta tesis.

1.1 Desplazamiento de objetos por comunidades de robots móviles (“Box-pushing problem”).

El problema de box-pushing se puede definir básicamente de la siguiente manera: dado un medio ambiente poligonal arbitrario, encontrar una ruta continua libre de colisiones que permita desplazar el objeto deseado desde un punto a otro, por medio de robots que sólo pueden empujarlo.

La tarea de transporte de objetos realizada por comunidades cooperativas de robots incluye entre varios problemas los de evitar colisiones, realizar manipulaciones estables, calcular trayectorias tanto para los robots como para el objeto, etc. En los casos más complejos los robots deberán manipular el objeto y cambiar no sólo su posición, sino también su orientación para poder desplazar el objeto en los lugares más estrechos del ambiente.

En general, el problema de box-pushing puede ser definido de la siguiente forma: teniendo un número N de robots móviles, mover la caja C hacia su objetivo T . El desplazamiento de C requerirá la cooperación de al menos un número N_c de robots móviles, entonces el problema de investigación fundamental en estos sistemas es el diseñar las leyes de control local de manera que se obtenga el comportamiento global deseado por la comunidad de robots (Figura 1.1).

Este problema tiene múltiples aplicaciones prácticas, entre las que destacan:

1.1. Desplazamiento de objetos por comunidades de robots móviles (“Box-pushing problem”).

4

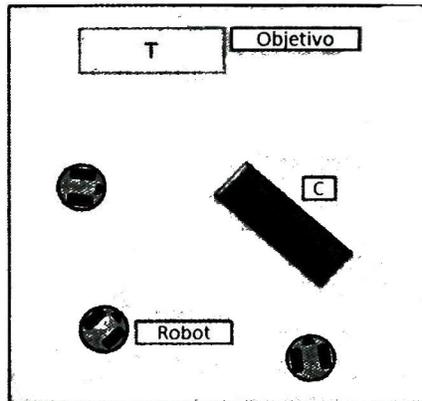


Figura 1.1: Esquema clásico del problema “box-pushing”.

- Rescate de personas.
- Realización de trabajos peligrosos.
- Automatización de procesos industriales.
- Realización de tareas de asistencia.
- Movimiento de objetos pesados.

Sin embargo, el aspecto más interesante de este problema es que representa un reto interesante para el desarrollo de esquemas de coordinación en sistemas multiagente. En efecto, el desplazamiento del objeto no puede realizarse si no existe un concierto entre los distintos integrantes de la comunidad de robots, de manera que la tarea pueda realizarse con éxito.

1.2 Planteamiento del problema

En esta tesis nos enfocaremos al cálculo de trayectorias de objetos que serán desplazados por comunidades de robots móviles en ambientes no estructurados². La aportación primordial de este trabajo es el desarrollo de 3 heurísticas³ originales que resuelven el problema encontrando rutas válidas para el objeto, tomando en cuenta algunos criterios para que dichas rutas permitan a los robots realizar la tarea de manera más precisa y rápida.

Para determinar las trayectorias del objeto, en este trabajo se considera que los robots móviles no son especializados en la manipulación, es decir, sólo pueden empujar la caja, mas no alzarla o tirar de ella. Sin embargo, es importante hacer notar que en la actualidad existen robots móviles, comercialmente disponibles, especializados en la manipulación de objetos [29].

Al utilizar robots no especializados, cada cambio de dirección durante el desplazamiento del objeto requiere que los robots modifiquen su posición con respecto a la caja para iniciar un nuevo movimiento (reconfiguración de los robots). Dichas reconfiguraciones requieren cierto tiempo en ser realizadas, por lo que si se logra reducir el número de reconfiguraciones se estará reduciendo el tiempo total de traslado del objeto.

Resulta entonces importante obtener trayectorias que además de ser solución al problema de desplazar el objeto de una posición inicial a una posición final deseada, el

²Un ambiente no estructurado será considerado como una área de trabajo compleja que contiene múltiples obstáculos.

³La palabra heurística proviene de la palabra griega heuriskein que significa descubrir, encontrar. Por heurística entendemos una estrategia, método o criterio que consta de un conjunto de pasos que deben realizarse para identificar, en el menor tiempo posible, una solución de alta calidad para un determinado problema difícil.

número de reconfiguraciones de los robots durante el movimiento sea lo más pequeño posible. Esto se logra encontrando trayectorias que minimicen el número de cambios de dirección de objeto, es decir, trayectorias que tengan una mayor 'continuidad'.

Con el problema del cálculo de la trayectoria del objeto resuelto, se plantean las bases para el análisis de los puntos de empuje, definidos como los puntos sobre el objeto en los que un robot puede apoyarse para empujar. Sin embargo, dependiendo de la orientación del objeto, sólo una parte de los puntos de empuje es coherente con la dirección de movimiento. Entre dos segmentos de la trayectoria, pueden existir puntos de empuje comunes que pueden ser explotados para reducir el número de robots que tienen que cambiar de posición durante una reconfiguración. Esta estrategia tendrá un impacto directo en el tiempo requerido para realizar las reconfiguraciones.

1.3 Objetivos del proyecto

El objetivo principal de este trabajo, es el desarrollo de una estrategia para realizar la planificación de trayectorias aplicada al problema de box-pushing en ambientes no estructurados, buscando reducir el tiempo, incrementar la precisión y calcular la mejor trayectoria libre de colisiones para el objeto.

De manera más específica, se pueden numerar los siguientes objetivos secundarios:

- Desarrollar una metodología que produzca soluciones al problema de planificación de trayectorias de un objeto en un medio ambiente no estructurado.
- Realizar un análisis del espacio de soluciones obtenido, con el objeto de encontrar la trayectoria solución que tenga el menor número de cambios de dirección. Con esto

se logra reducir el número de reconfiguraciones de los robots al rededor del objeto.

- Establecer las bases del análisis de los puntos de empuje en las rutas obtenidas.

1.4 Breve descripción del enfoque propuesto

La estrategia propuesta para resolver el problema del box-pushing se compone de 5 etapas básicas (Figura 1.2).

El primer paso consiste en realizar una descripción semialgebraica del medio ambiente. Esto se logra por medio de la representación de cada objeto del medio ambiente como un conjunto de inecuaciones, de modo que el entorno queda representado por polígonos convexos. Esta misma representación se aplica al objeto a desplazar.

Con el conjunto de restricciones que representan al objeto y a los obstáculos, el segundo paso consiste en implementar un mecanismo de detección de colisiones que nos permitirá construir el espacio de configuraciones libres, es decir, el conjunto de posiciones en el espacio en el que el objeto puede situarse sin estar en contacto con algún otro objeto del medio ambiente. El resultado de esta etapa es una representación matricial del espacio de configuraciones.

En la tercera etapa, se aplica un algoritmo de frente de onda en la matriz del espacio de configuraciones, con lo cual se obtiene el espacio de trayectorias solución al problema de planificación de trayectorias sin colisión.

La construcción formal de las trayectorias del objeto se realiza en el cuarto paso. Aquí se aplica una metodología heurística de optimización, para encontrar la trayectoria

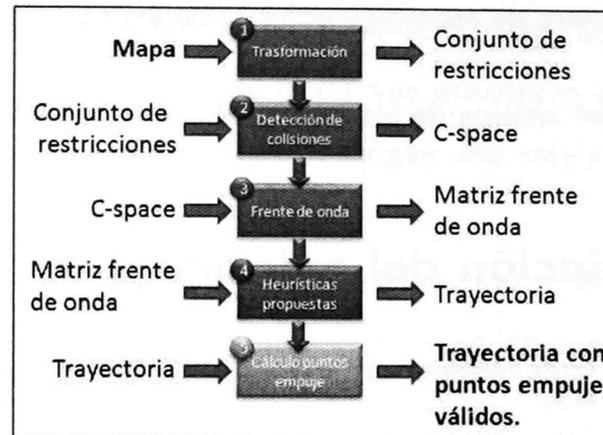


Figura 1.2: Esquema básico de la estrategia propuesta.

que requiera el menor número de cambios de dirección, con el objeto de reducir el número de maniobras de los robots alrededor del objeto. Particularmente, se exploraron diversas heurísticas para encontrar aquella que ofreciera el mejor compromiso tiempo-calidad de la solución.

Finalmente y concluido el paso cuatro, la siguiente acción es determinar los puntos de empuje válidos para cada segmento de la trayectoria obtenida, identificando los puntos de empuje y descartando los puntos de arrastre.

Las principales características de la solución propuesta se listan a continuación.

- Obtener las trayectorias más cortas.
- Buscar trayectorias con un menor número de reconfiguraciones de los robots (manejaremos el término "reconfiguraciones" para indicar los cambios de posición de los robots alrededor del objeto, de modo que estos puedan empujar adecuadamente la caja).
- Dar preferencia a ciertas rotaciones de la caja, con el fin de mejorar los puntos de

empuje.

1.5 Conjunto de hipótesis iniciales

- Los robots deberán operar en espacios de trabajo conocidos previamente, delimitados y con obstáculos dispuestos arbitrariamente (pero cuya posición es conocida inicialmente).
- El objeto a desplazar, así como los obstáculos en el medio ambiente podrán tener una forma arbitraria, con la restricción de que puedan ser representados por un conjunto de polígonos convexos (en el caso de los polígonos cóncavos, bastará con representarlos con más de un polígono convexo).
- Los robots involucrados en las tareas de empuje sólo serán capaces de empujar la caja y serán de tipo no especializado, por lo tanto no se consideran movimientos en los que se sujete el objeto para levantarlo o tirar de él. Por sus dimensiones, sólo se requiere que todos los robots del equipo de trabajo tengan dimensiones homogéneas.
- Es necesario incluir las dimensiones de los robots en el cálculo de la trayectoria, para permitir que estos puedan circular alrededor del objeto en cualquier posición considerada "libre". El cálculo del espacio de configuraciones se hará considerando el objeto al cual se le habrá aplicado previamente un incremento al tamaño equivalente a la dimensión de los robots.
- Se modelarán las configuraciones de la caja en un espacio discreto, por lo que sólo se obtendrán soluciones dentro de una cierta resolución definida a priori.
- Sólo se resolverá el problema geométrico (solución considerando únicamente la posición instantánea de los robots y de los objetos) y se dejarán para trabajos

futuros los aspectos cinemático y dinámico (velocidades, aceleraciones, inercias, etc.).

- El conjunto de cálculos será centralizado y fuera de línea.
- Los puntos de contacto entre los robots y el objeto estarán definidos a priori y sólo servirán como puntos de empuje.

Lo que se pretende obtener es una planificación automática de la trayectoria, determinando los movimientos necesarios para trasladar el objeto evitando colisionar con los obstáculos presentes en dicho ambiente.

1.6 Estructura del reporte de tesis

El resto de este documento se organiza de la siguiente forma: en el capítulo 2 se presenta una breve reseña de la historia de la robótica y presenta una revisión del estado del arte del problema; en el capítulo 3 se describe la primera fase del proyecto, esta parte es enfocada a la representación del objeto y del espacio de trabajo. La segunda fase del proyecto relacionada al cálculo de trayectorias y la definición formal de las herramientas utilizadas para construir la estrategia serán presentadas en la sección 4. La descripción de los casos de pruebas y la presentación de los resultados obtenidos se realizará en el capítulo 5. Finalmente, las conclusiones y perspectivas serán presentadas en la sección 6.

2

Estado del arte

“Esperamos que en el año 2013 haya un robot en cada hogar de Corea.”

Ho-Gil Lee¹

En este capítulo se presentará un breve resumen de la historia de la robótica móvil; se definirá formalmente el problema del movimiento de objetos por robots (box-pushing) y finalmente se presentará una revisión bibliográfica de los métodos más representativos propuestos para la resolución de este problema.

2.1 Historia de la robótica móvil

Dos grandes deseos humanos pueden considerarse como los impulsores de la robótica: el crear entes artificiales que simularan la vida y, aún más antiguo, el descargarse del trabajo difícil.

¹Ho-Gil Lee es el director del Korean Institute of Industrial Technology (KITECH) [13].

La palabra 'robot' proviene de la palabra checa 'robota', que significa "labor forzada", servicio, esclavo, etc. Este nombre fue utilizado en el imperio austro-húngaro hasta 1848. Fue introducida en la literatura en el año de 1920 en la obra 'R.U.R' (Rossum's Universal Robots) por el escritor checo Karel Capek. En realidad, la invención de la palabra se debe a su hermano, Josef, mientras que Karel la utilizó en su obra.

No existe una definición única de qué es un robot. En general, un robot, para ser considerado como tal, debe presentar algunas de las siguientes propiedades:

- No es natural, sino que ha sido creado artificialmente.
- Puede sentir su entorno.
- Puede manipular cosas de su entorno.
- Tiene cierta inteligencia o habilidad para tomar decisiones basadas en el ambiente o en una secuencia preprogramada automática.
- Es programable.
- Puede moverse en uno o más ejes de rotación o traslación.
- Puede realizar movimientos coordinados.

El termino Robot según la Enciclopedia Británica tiene el siguiente significado: Máquina operada automáticamente que sustituye el esfuerzo de los humanos, aunque no tiene por qué tener apariencia humana o desarrollar sus actividades de manera similar a la de los humanos.

Por otra parte, el Diccionario Merriam Webster define el término robot como: Máquina que se asemeja a los humanos y desarrolla como ellos tareas complejas como

andar o hablar. Un dispositivo que realiza automáticamente tareas complicadas.

Robot, según el Diccionario de la Real Academia de la lengua Española es una Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas.

A pesar de las múltiples definiciones existentes, aún no hay acuerdos formales acerca de la definición precisa de lo que se considera un robot. Joseph Engelberger, un pionero en la industria robótica, expresó claramente esta idea con su frase: "No puedo definir un robot, pero reconozco uno cuando lo veo".

La evolución de la robótica dividió a esta misma en diferentes vertientes basadas en las capacidades los robots. Por ejemplo, robots industriales, humanoides, móviles, etc.

La robótica móvil debe su existencia a la evolución de la robótica hacia los sistemas móviles autónomos. Este tipo de robots se caracterizan por su capacidad de traslado en entornos dinámicos y desconocidos sin necesidad de supervisión.

William Grey Walter creó los primeros robots móviles autónomos de la historia. Los robots ELSIE (Electro-Light-Sensitive Internal-External) y ELMER (Electro-MEchanical Robot) fueron construidos en Inglaterra en 1948 y 1949 y poseían muy limitadas capacidades de desempeño. Estos robots basaban su navegación en un sistema mecánico que le permitía seguir una fuente de luz y otro que le permitía detectar el contacto con objetos.

En 1954 Devol diseñó el primer robot programable y acuñó el término "autómata

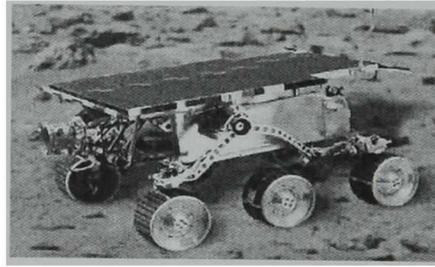


Figura 2.1: Mars-Rover.

universal", que posteriormente recorta a Unimation y registra una patente llamada 'Programmed article transfer' (Transferencia de artículos programada). Engleberger llamaría también 'Unimation' a la primera compañía dedicada a la construcción de robots. Cuatro años después Unimation lanza al mercado su primer robot industrial, que utilizaba los principios de control numérico en un robot manipulador de transmisión hidráulica.

En 1968 se dio a conocer el robot Shakey del SRI (Stanford Research Institute). Este robot estaba equipado con una serie de sensores y una cámara encargada de la visión, y podía desplazarse libremente por el suelo, explorar y representar el entorno mediante un mapa de tipo rejilla. Una de las grandes aportaciones de este robot fue que, para realizar el procesamiento de la información, se implementaron 2 procesadores, uno montado en el robot (para el control de los sensores y motores) y otro externo que permitía realizar el procesamiento de las imágenes. La comunicación entre estos procesadores se realizaba por radio.

En los 70, la NASA junto con el "Jet Propulsion Laboratory" iniciaron un programa que incluía el desarrollo de plataformas capaces de explorar terrenos hostiles. El primer fruto de este programa sería el MARS-ROVER, equipado con un brazo mecánico de tipo Stanford, un dispositivo telemétrico láser, cámaras estéreo y sensores de proximidad (Figura 2.1).

A inicios de los 80 el equipo de Ernest Dickmanns en la Universidad de Bundeswehr construyó el primer automóvil robot. El vehículo desarrollado era capaz de circular a una velocidad de 55mph en calles vacías. Otro desarrollo interesante de la década fue el robot CART, que tenía la capacidad de procesar imágenes estéreo. Este robot era capaz de modelar los obstáculos mediante las coordenadas cartesianas de sus vértices y por lo tanto era capaz de realizar cálculos de trayectorias de manera más eficiente que con el uso de rejillas. Sin embargo, el simple hecho de cruzar una habitación le tomaba en promedio 5 horas, debido a que las exigencias de cómputo eran muy elevadas y las capacidades de procesamiento muy pobres.

Otro robot móvil desarrollado en la década de los 80 fue el CMU-ROVER de la Universidad Carnegie Mellon. Su más importante aportación fue la incorporación de una rueda timón, lo que le permitía adoptar cualquier posición y orientación en el plano. Para su control, este robot implementaba un sistema de control jerárquico apoyado por una rudimentaria memoria compartida que le permitía reducir los tiempos de procesamiento de manera importante.

En 1991 nacen los robots Khepera. Sus diseñadores Edo Franzi, André Guignard y Francesco Mondada diseñaron este pequeño robot con el fin de apoyar las actividades de investigación de la robótica móvil. Actualmente es un robot muy utilizado en investigación de sistemas multiagente.

El robot programable Pioneer inicia su comercialización en 1995. La empresa que lo produce es "Mobile Robots" y su precio asequible lo convierte en un referente para la educación e investigación (Figura 2.2).



Figura 2.2: Robot Pioneer P3-DX.

En 1996 la NASA envía a Marte la misión Mars Pathfinder con su robot móvil "Sojourner", diseñado para explorar la superficie marciana comandado desde la Tierra.

A finales de 1998, la empresa LEGO introdujo la serie de juguetes educativos MindStorm (RIS 1.0, Robotics Invention System), con el cerebro RXC (Robotics Command eXplorer) capaz de controlar 3 motores, 3 sensores y un puerto infrarrojo para comunicación serial. Tiempo después esta plataforma evolucionaría hasta la generación NXT, con un nuevo esquema de comunicación basada en el estándar Bluetooth y una nueva gama de sensores (Figura 2.3).

El inicio de la era de los robots caseros la marca Sony con la introducción en 1999 de "Aibo", un robot con forma de perro que es capaz de ver, caminar e interactuar con el ambiente. Tres años después, la empresa iROBOT inicia la comercialización de "Romba", un robot móvil doméstico que puede desplazarse dentro una casa para limpiar los pisos.

En la actualidad, la robótica se debate entre modelos sumamente ambiciosos como

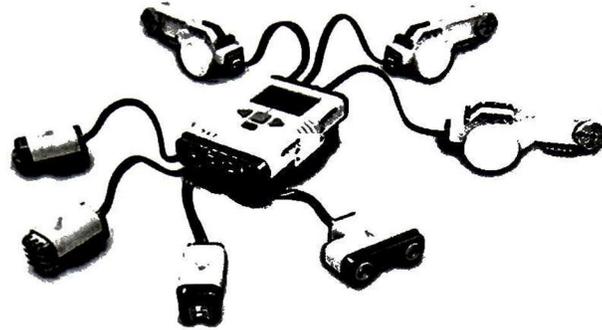


Figura 2.3: Robot Lego NXT.

el CYPHER, un helicóptero robot de uso militar, robots anfibios capaces desplazarse en tierra y agua, aviones sin piloto, etc.

2.2 Tipo de robots y de movilidad

“Quizás el aspecto más importante en el diseño de la estructura física de una criatura artificial es la selección del mecanismo de locomoción” (Brooks & Ferrell, 1996)

Una de las características primordiales de los robots móviles es su mecanismo de locomoción. Es claro que la selección de mecanismos de locomoción estará directamente relacionada con la tarea que se desee realizar y el ambiente en el que se desplazarán los robots.

En la actualidad los robots móviles se sirven de diferentes mecanismos de locomoción. Díaz [3] define las siguientes clasificaciones:

1. Robots móviles a ruedas
2. Robots móviles a patas

3. Robots móviles híbridos
4. Robots móviles voladores

2.2.1 Robots móviles a ruedas

La rueda es un método altamente probado, que presenta una gran cantidad de ventajas y es muy eficiente en superficies homogéneas poco inclinadas.

Ventajas:

- Eficiente en cuanto a consumo de energía.
- Tecnología muy conocida y probada.
- Mecánicamente sencillo y muy robusto.

Desventajas:

- Adaptados para superficies lisas poco inclinadas.
- Presentan restricciones no holonómicas², es decir, no son omnidireccionales, ya que existen direcciones sobre las cuales no puede desplazarse sin reorientación.

2.2.2 Robots móviles a patas

Estos son, evidentemente, mecanismos inspirados del enfoque biológico. Sus ventajas, entre otras, son la adaptabilidad al terreno y la posibilidad de omnidireccionalidad. Se encuentra una gran variedad de diseños y aplicaciones, algunas comerciales. Por ejemplo: los robots bípedos QRIO de Sony [23], Asimo de Honda [9], el robot con forma de insecto

²Conjunto de restricciones físicas del movimiento de los robots móviles en las que interviene la velocidad. Por ejemplo, la condición de rodadura sin deslizamiento de una rueda en un plano.

Genghis desarrollo por Rodney Brooks en 1989 [1] y el famoso robot con forma de perro Aibo de Sony [22] entre otros.

Ventajas:

- Pueden, en principio, moverse por mayor variedad de terrenos, incluso discontinuos (escaleras).
- Permiten el desplazamiento omnidireccional.

Desventajas:

- Mayor consumo de energía.
- Mayor complejidad mecánica y de control.

2.2.3 Robots móviles híbridos

Algunos métodos utilizan combinaciones entre ruedas y patas, otros son adaptaciones que han resultado convenientes para algún tipo de terreno o alguna aplicación, por ejemplo los robots con orugas.

Ventajas:

- Por la especialización de sus elementos son muy eficientes en los terrenos para los que fueron diseñados.

Desventajas:

- Mayor complejidad.
- Por su alta especialización, no son aptos para cualquier tipo de terreno.

2.2.4 Robots móviles voladores

Se trata de robots de un tipo especial ya que suelen ser robots teleoperados más orientados a la aeronáutica que a la robótica.

Ventajas:

- No son afectados por el tipo de terreno.

Desventajas:

- Control complejo.

2.3 Problemas asociados a la robótica móvil

El desarrollo de la robótica móvil trajo consigo una serie de problemas inherentes a la movilidad. En primer lugar, el problema de determinar la posición de un robot en su entorno usando sus propios sensores, denominado "Problema de localización". La localización de un robot en su entorno es uno de los problemas clásicos de la robótica móvil y es el punto de partida de otros problemas como la interacción con el entorno, y sobre todo, la navegación. La mayor parte de las tareas que un robot móvil tiene que realizar depende de su posición en el mismo.

Otro problema relacionado a la robótica móvil es la generación y mantenimiento de mapas, que representa una área de investigación altamente activa en robótica móvil y en inteligencia artificial (IA). La generación de mapas aborda el problema de adquirir los modelos espaciales de ambientes físicos a través de los robots móviles y es generalmente reconocido como uno de los desafíos más importantes de la robótica móvil autónoma. En la actualidad, existen métodos robustos para trazar ambientes estáticos, estructurados, y

de tamaño limitado. A pesar del progreso significativo en esta área, todavía se plantean grandes desafíos. Los ambientes no estructurados, dinámicos, o de dimensiones muy elevadas siguen siendo un gran problema abierto a la investigación.

Finalmente, encontramos el problema de planeación de trayectorias, al que pertenece el trabajo propuesto en esta tesis. En el ámbito de la robótica, la planeación se entiende como la automatización de los sistemas mecánicos que tienen capacidad de sentir, actuar y que además tienen capacidades de cómputo (términos tales como sistemas autónomos o agentes también se utilizan). Una necesidad fundamental en robótica es contar con algoritmos que conviertan tareas especificadas en alto nivel por los seres humanos a descripciones de bajo nivel de cómo ejecutarlas. Los términos planeación de trayectorias y planeación de movimientos son términos también usados para esta clase de problemas. Una versión clásica de este tipo de problemas es el problema del movimiento del piano.

2.4 Problema del movimiento del piano (piano mover's problem)

El problema de "*box-pushing*" mencionado anteriormente, está estrechamente relacionado al bien conocido problema del movimiento del piano ("*Piano's mover problem*") descrito por Schwartz y Sharir en 1983. La formulación clásica del problema del movimiento del piano se define de la siguiente manera: dado un objeto poliédrico rígido arbitrario P , y un ambiente poliédrico, encontrar una trayectoria continua y libre de colisión para llevar a P desde una cierta configuración inicial a una configuración final deseada (Figura 2.4).

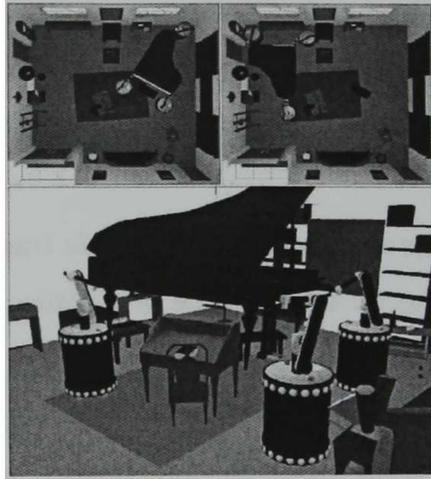


Figura 2.4: Movimiento de un piano usando robots móviles.

De manera más formal, en WolframMathWorld [27] el problema del movimiento del piano se define como: Dado un subconjunto abierto v en un espacio n -dimensional y dos subconjuntos compactos C_0 y C_1 de v , de donde C_1 se deriva C_0 por un movimiento continuo, ¿es posible moverse de C_0 a C_1 mientras que se permanece enteramente dentro de v ?

Según LaValle [12] la formulación del problema del movimiento del piano se da de la siguiente manera:

Supongamos que el mundo, $W = \mathbb{R}^2$ o $W = \mathbb{R}^3$, contiene una región de obstáculos, $O \subset W$ y un robot rígido, $A \subset W$. Se asume que A y O están expresados como modelos semi-algebraicos (que incluyen modelos poligonales y poliédricos). A partir de esto, se define el espacio de configuraciones C de A como, $q \in C$, con $q = (x_t, y_t, \theta)$ para $W = \mathbb{R}^2$ y $q = (x_t, y_t, z_t, h)$ para $W = \mathbb{R}^3$ (h representada en cuaterniones). El conjunto C representa entonces todas las posibles configuraciones que el robot A puede tomar dentro de W .

La región de obstáculos C_{obs} es un subconjunto del espacio de configuraciones C definido como:

$$C_{obs} = \{q \in C \mid A(q) \cap O \neq \emptyset\},$$

donde $A(q)$ representa la transformación del robot A para llevarlo a la configuración q . El subconjunto C_{obs} representa entonces todas las configuraciones donde A intersecta la región de obstáculos, O .

Las configuraciones restantes se consideran como espacio libre y se denotan como $C_{libre} = C \setminus C_{obs}$.

A partir de todos estos elementos se puede definir formalmente el problema de planificación de trayectorias.

Formulación del problema del movimiento del piano

1. Dado un mundo W definido en $W = \mathbb{R}^2$ o $W = \mathbb{R}^3$.
2. Una *región de obstáculos* semi-algebraica $O \subset W$ en el mundo.
3. Un *robot* semi-algebraico definido en W . Puede ser un robot rígido A o una colección de los m acoplamientos, A_1, A_2, \dots, A_m .
4. El *espacio de configuraciones* C ("C-space") determinado por el sistema de todas las transformaciones posibles que se pueden aplicar al robot. De esto, C_{obs} y C_{libre} son derivados.
5. Una configuración $q_I \in C_{libre}$ designada como configuración inicial.

6. Una configuración $q_F \in C_{libre}$ definida como *configuración final o meta*. Las configuraciones inicial y final juntas son llamadas a menudo como "*Query pair*" (o "*query*") y se señalan como (q_I, q_F) .
7. Un algoritmo completo debe computar la trayectoria (continua), $\tau : [0, 1] \rightarrow C_{libre}$, tal que el $\tau(0) = q_I$ y $\tau(1) = q_F$, o que reporte correctamente que no existe tal trayectoria.

Fue demostrado por Reif [19, 20] que este problema es de tipo PSPACE-duro, que implica NP-duro. La principal dificultad radica en que no existen métodos directos ni eficientes para determinar los límites exactos o la representación completa de C_{libre} o de C_{obs} . El mayor problema es que la dimensión de C es ilimitada, y depende únicamente del número de movibilidades del robot.

2.5 Revisión bibliográfica

Esta área de investigación tiene una amplia gama de aplicaciones prácticas, que la hacen particularmente interesante para ser estudiada. En esta sección, se describen brevemente algunos de los trabajos relacionados más destacables que han sido reportados. En la sub-sección 2.5.1 se agrupan algunos de los múltiples esquemas propuestos para resolver el problema del box-pushing, pero que no se pueden comparar directamente con el esquema propuesto en esta tesis. En la sub-sección 2.5.2, se reportan los trabajos más similares al presentado en este documento.

2.5.1 Esquemas existentes

Existen múltiples estrategias propuestas para resolver el problema del movimiento de objetos por comunidades de robots móviles. Las diferencias entre ellas pueden ser

mayúsculas: en algunos casos el procesamiento es centralizado (es decir que un ente realiza todo el procesamiento y comunica al resto del equipo las acciones que se deben realizar), en otros es descentralizado (cada robot realiza el cálculo de sus acciones). Otras variantes que encontramos entre los mecanismos propuestos enfocan el hardware que utilizan los robots, por ejemplo: algunos utilizan cámaras de vídeo para localizar y controlar el movimiento de los objetos y otros sólo basan su funcionamiento en la utilización de sonares y de contacto. Finalmente la cantidad y el tipo de los robots es un punto que varía entre cada mecanismo, siendo los equipos de robots homogéneos los más utilizados.

La comunicación es un factor esencial en los sistemas multi-robot. Yamada [28] en sus estudios reporta que, independientemente de la configuración del sistema (centralizado o descentralizado), se pueden omitir las necesidades de comunicación y substituir estas carencias con la aplicación de mecanismos de comportamiento basados en información local. Estas estrategias no varían de forma dinámica y es necesario tener una amplia gama de comportamientos pre-programados para resolver nuevos retos. En contraste, Muñoz [17] reporta cómo la comunicación puede mejorar considerablemente el funcionamiento de los sistemas multi-robot, aumentando la coordinación, la cooperación, realizando el manejo de colisiones y finalmente abriendo la posibilidad de implementar esquemas cliente-servidor entre los robots involucrados.

Existen estudios que enfocan el problema mediante una estrategia utilizada en robots manipuladores conocida como "Object closure" (Figura 2.5). Wang [26] reporta que la manipulación de objetos puede ser realizada rodeando el objeto por un conjunto de robots, por lo que la posición del objeto estará controlada por la posición de cada robot que rodea y a su vez empuja el objeto. Sin embargo, esta teoría fundamenta su

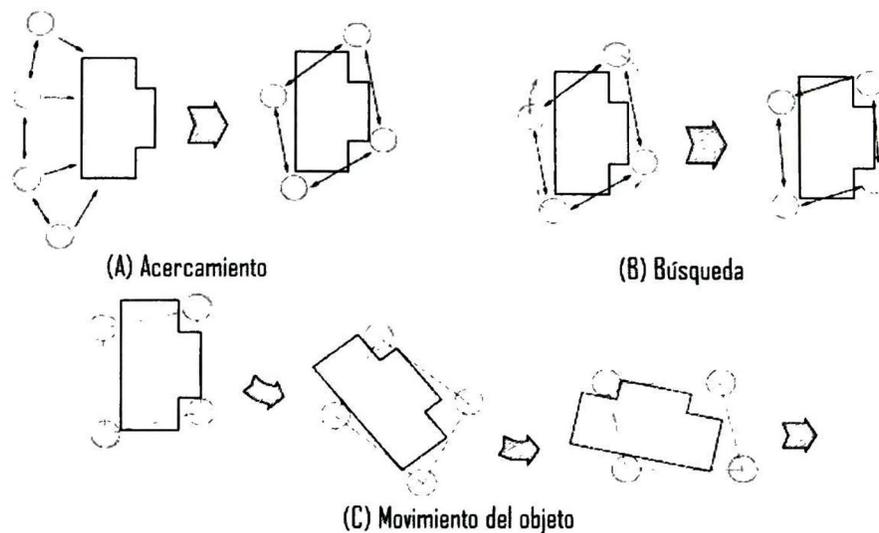


Figura 2.5: "Object closure"

funcionamiento en robots holonómicos y en el uso de una gran cantidad de robots móviles.

Otra estrategia de solución propuesta es conocida como "Pusher-Watcher" (Figura 2.6) en la que un robot "vigía" observa el movimiento del objeto y coordina las acciones del equipo de robots que manipulan la caja. Gerkey [7] reporta un modelo basado en este esquema, utilizando modelos de aprendizaje reforzado. Este trabajo muestra buenos resultados, pero utiliza robots muy especializados y se auxilia de técnicas de visión, mismas que no pueden ser aplicadas en la mayoría de los simuladores disponibles. Por otro lado, Kovac [11] trabaja con el mismo esquema pero utiliza robots holonómicos y sus pruebas sólo fueron realizadas en simulador.

Otros trabajos se enfocan en el uso de modelos "swarm intelligence" (Figura 2.7). Li [14] experimenta usando comunidades de robots homogéneos. Estos modelos se inspiran principalmente de las comunidades de insectos e investigan el modelo de coordinación de agentes simples. Este enfoque ofrece cuatro características principales:

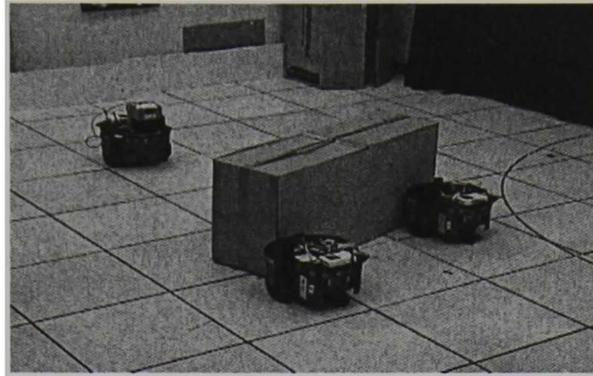


Figura 2.6: "Pusher-Watcher"

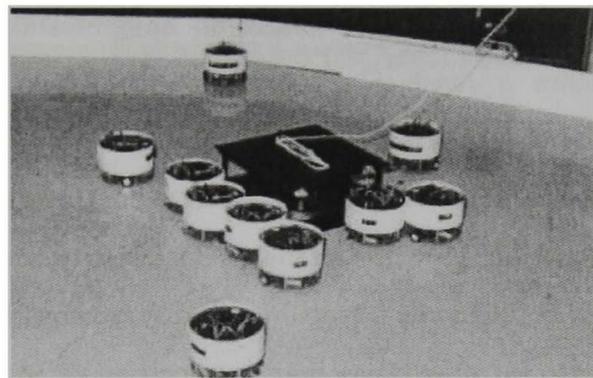


Figura 2.7: "swarm intelligence"

descentralización, homogeneidad, auto organización y simplicidad. Aún cuando reporta buenos resultados, este trabajo no se ha llevado a cabo con implementaciones reales.

Otra de las estrategias más avanzadas reportadas hasta hoy es el aprendizaje reforzado. Wang [25] implementa una variante del aprendizaje reforzado que incluye un mecanismo de decisión basado en un proceso de Markov, también conocido como "Q-Learning". Las capacidades de almacenamiento y procesamiento del aprendizaje pueden llegar a ser demasiado importantes, además de que un proceso de aprendizaje amplio es necesario para obtener el modelo final.

2.5.2 Esquemas basados en el uso de espacios de configuraciones

El trabajo desarrollado en esta tesis se basa en una búsqueda de soluciones en una representación discreta del espacio de configuraciones. Existen muy pocos trabajos que utilicen este tipo de enfoque.

Entre las estrategias que solucionan el problema en ambientes no estructurados y enfocados específicamente a resolver el problema geométrico se encuentran los métodos propuestos por Gene [6] y Lengyel [10]. Ambos autores parten de un mapa conocido y tratan de resolver el problema del cálculo de trayectorias para la caja, utilizando una representación del espacio basado en la construcción de un espacio de configuraciones (C-space) y el cálculo de trayectorias lo realizan con el algoritmo de frente de onda convencional. Entre las diferencias más sustanciales están que Lengyel [10] propone un método para el cálculo de colisiones basado en el procesamiento de imágenes con hardware especializado, lo que definitivamente implica un costo elevado de procesamiento. Gene [6] utiliza mapas de celdas para representar el ambiente, en el que cada celda almacena 7 parámetros que le son de utilidad para calcular la trayectoria.

Un ejemplo de las trayectorias obtenidas por estos dos autores son mostradas en las Figuras 2.8 y 2.9, y es destacable que ambos autores utilizan un mecanismo para calcular trayectorias basados en un algoritmo de frente de onda simple, no definen el tipo de robots para los que funcionan las trayectorias obtenidas y no hacen ningún análisis de los puntos de empuje.

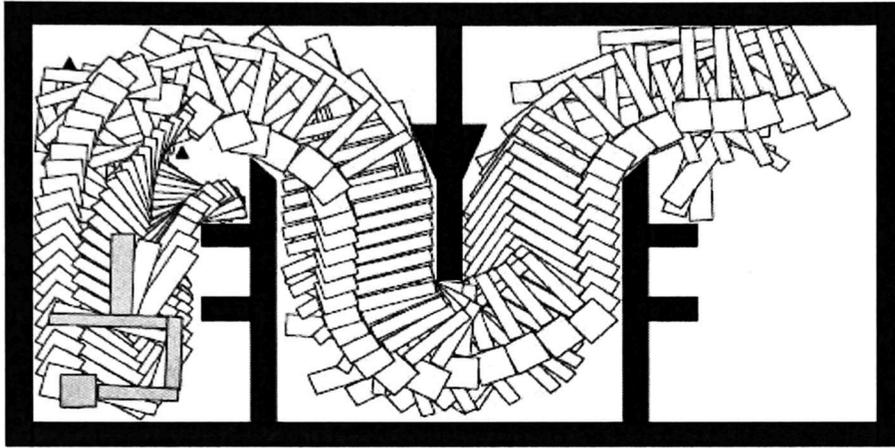


Figura 2.8: Ejemplo de trayectoria

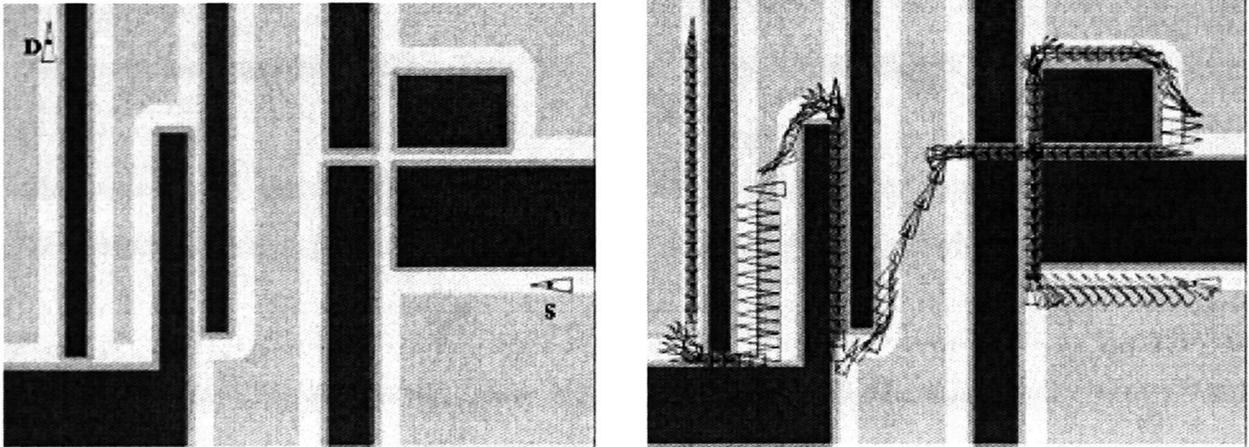


Figura 2.9: Ejemplo de trayectoria

2.6 Resumen del capítulo

Como se mostró en este capítulo, desde su inicios la robótica ha presentado grandes retos para los investigadores, uno de ellos es el problema del movimiento de objetos realizado por comunidades de robots móviles no especializados, este problema ha captado el interés de muchos investigadores y en la actualidad se han publicado múltiples estrategias que intentan resolverlo (por ejemplo: "Object closure", "Pusher-Watcher", "swarm intelligence", etc.), sin embargo, la estrategia que se presenta en este trabajo además de resolver el problema del cálculo de trayectorias, se enfoca a reducir el número de cambios de dirección de la caja, con lo que se pretende tener mejoras en los tiempos de traslado y aumentar la precisión de los movimientos.

3

Representación del espacio

“La era de los robots será en las primeras décadas del siglo XXI”

“Esperamos que eventualmente la industria de la robótica crezca mas rápido que la industria de las computadoras” Toshi T. Doi¹

Al trabajar con una propuesta de cálculo de trayectorias global, es necesario identificar el espacio libre de ambiente, encontrar las posiciones en las que el objeto en desplazamiento se encuentra en colisión con los obstáculos e identificar los obstáculos en el ambiente para poder evitarlos. Por supuesto esta serie de cálculos deben de ser almacenados y procesados adecuadamente para que los robots puedan realizarlos automáticamente.

En este capítulo se describe la primera fase en la construcción de la propuesta de solución, los problemas que se resolverán son: la transformación del mapa descrito por

¹Toshi T. Doi es el presidente del laboratorio de criaturas digitales de Sony. Declaración hecha en el marco de la exposición Robodex2000.

coordenadas a un sistema de inecuaciones, el proceso de cálculo de colisiones y con este último construir un espacio de configuraciones discreto.

3.1 Transformaciones del espacio

Como se definió en el capítulo anterior, el espacio de configuraciones (C-space) C es el conjunto de todas las posibles posiciones q que puede tomar un robot rígido A (en nuestro caso el objeto a desplazar) en un medio ambiente plano $W = \mathbb{R}^2$ con un conjunto de obstáculos fijos O .

Para realizar la planificación de trayectorias, es necesario determinar cuáles configuraciones q en C implican una colisión de A con O (y que no pueden ser utilizadas) y en cuáles no existe intersección entre A y los obstáculos, obteniendo así una descripción de C_{obs} y de C_{libre} .

En nuestro caso, tanto los obstáculos O como el objeto A están representados por las coordenadas de los vértices que los delimitan: O está definido con respecto al sistema coordenado del medio ambiente W , mientras que los vértices de A están definidos con respecto al centro de masa del objeto.

En la Figura 3.1 se muestra un ejemplo del modelado poligonal de un ambiente real que consta de 17 obstáculos. La Figura 3.2 muestra un ejemplo de un objeto A que es modelado poligonalmente en su propio sistema coordenado. Finalmente, en la Figura 3.3 se muestra el conjunto de obstáculos y el objeto en el mismo sistema coordenado.

Para determinar si existe colisión en la configuración $q = (x_t, y_t, \theta)$, es necesario

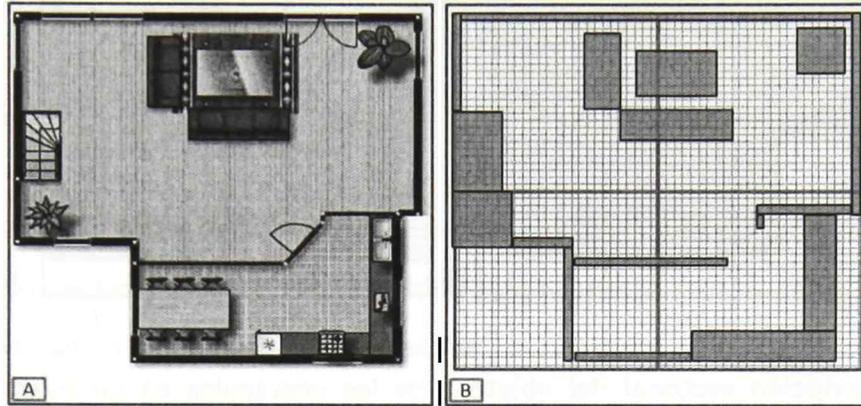


Figura 3.1: Descripción vectorial del mundo en su sistema coordenado.

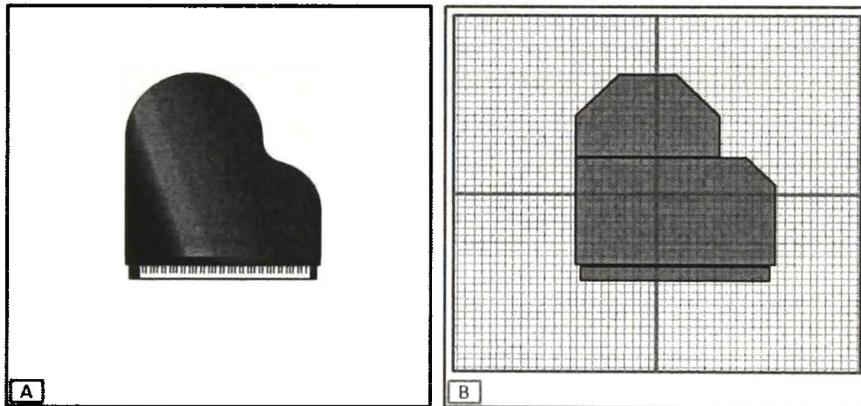


Figura 3.2: Descripción vectorial del objeto en su sistema coordenado.

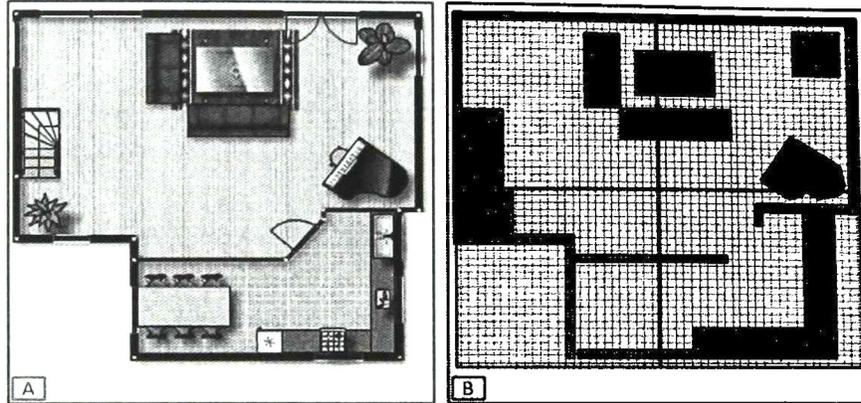


Figura 3.3: Descripción vectorial del objeto y de los obstáculos en un mismo sistema coordenado.

calcular primero la representación del objeto A en esa posición, que denominaremos $A(q)$. Esto se logra aplicando una transformación lineal a las coordenadas de los vértices de A , en función de los valores de q , como se muestra en el algoritmo 2.

Algoritmo 1 Transformación lineal

Entrada: Configuración q deseada del objeto A

Salida: Coordenadas (x, y) del objeto A en la configuración deseada q

$Cci[]$ ← Coordenadas (x_i, y_i) del objeto A en su configuración inicial i

$Cct[]$ ← Coordenadas (x_t, y_t) del objeto A en la configuración deseada q

para todos los vértices de A **hacer**

$$Ccf[] \leftarrow \begin{bmatrix} \cos \theta & -\sin \theta & x_t & x_i \\ \sin \theta & \cos \theta & y_t & y_i \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

fin para

regresa $Ccf[]$

Para determinar cuáles configuraciones pertenecen al espacio libre, utilizaremos un algoritmo de detección de intersecciones entre polígonos convexos. Por ello, es necesario expresar cada obstáculo del medio ambiente y el objeto en la configuración deseada como representaciones semialgebraicas, esto es, como conjuntos de restricciones lineales. Así,

cada polígono en consideración quedará representado de la siguiente forma:

$$\vec{n}_i \cdot X = a_i x + b_i y \leq d_i, i = 1 \dots P_k$$

En donde P_k es el número de caras del objeto k y para cada (a_i, b_i) ; $a_i \neq 0$ or $b_i \neq 0$

Entonces el primer paso en la construcción del espacio de configuraciones es la transformación de los polígonos (representado originalmente por las coordenadas de sus vértices) a un conjunto de inecuaciones lineales.

Para obtener la descripción semialgebraica de los objetos, consideramos que los puntos están dados en orden trigonométrico, como se muestra en la Figura 3.4. Los pasos para obtener el conjunto de inecuaciones de cada polígono son los siguientes.

1. Obtener el vector director entre cada par de puntos consecutivos.

$$\vec{u} = AB = X_b - X_a$$

2. Partiendo del vector director, encontrar su vector perpendicular.

$$\vec{m} = R \vec{u} \text{ en donde } R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \text{ con } \theta = 90$$

3. Obtener el vector normal externo.

$$\vec{n}_i = \frac{1}{\|\vec{m}\|} \cdot \vec{m}$$

4. Finalmente calcular la distancia existente entre el vector normal y el origen.

$$d_i = \vec{n}_i \cdot \vec{x}_a = \vec{n}_i \cdot \vec{x}_b$$

5. Los pasos 1 al 4 serán repetidos para cada polígono que represente tanto al objeto como al obstáculo.

En el algoritmo 2 se muestra el pseudocódigo del proceso utilizado para calcular el conjunto de inecuaciones con las que se representará el ambiente.

Algoritmo 2 Calcula inecuaciones

Entrada: Coordenadas (x, y) de un polígono p
Salida: Conjunto de inecuaciones que definen al polígono p
 $Cobj[x_{obj}][y_{obj}] \leftarrow$ Coordenadas del polígono p
 $Vectores[][] \leftarrow \emptyset$
 $Normas[] \leftarrow \emptyset$
 $Distancias[] \leftarrow \emptyset$
 $lados \leftarrow$ número de caras del polígono p
para $k = 1$ hasta $k = lados$ **hacer**
 para $j = 1$ hasta $j = 2$ **hacer**
 si $k = lados$ **entonces**
 $Vectores[k][j] \leftarrow Cobjs[k + 1][j] - Cobjs[k][j]$
 sino
 $Vectores[k][j] \leftarrow Cobjs[0][j] - Cobjs[k][j]$
 fin si
 fin para
fin para
para $l = 1$ hasta $l = lados$ **hacer**
 $Vectores[l] \leftarrow Vectores[l] \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$
fin para
para $m = 1$ hasta $m = lados$ **hacer**
 $Normas[m] \leftarrow \sqrt{Vectores[m][0]^2 + Vectores[m][1]^2}$
 para $n = 1$ hasta $n = 2$ **hacer**
 $Vectores[m][n] \leftarrow \frac{Vectores[m][n]}{Normas[m]}$
 fin para
 para $q = 1$ hasta $q = lados$ **hacer**
 $Distancias[q] \leftarrow Vectores[q][0] \cdot Cobjs[q][0] + Vectores[q][1] \cdot Cobjs[q][1]$
 fin para
fin para
regresa $Vectores[][]$, $Distancias[]$

Al término de este proceso, se cuenta con el conjunto de restricciones (Figura 3.4) que describen tanto nuestro ambiente como al objeto en la configuración deseada.

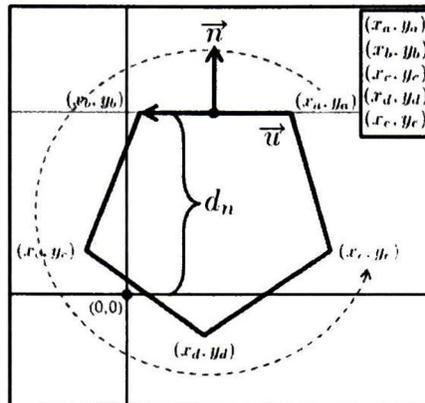


Figura 3.4: Representación gráfica de los vectores que definen las restricciones.

3.2 Intersecciones entre polígonos (colisiones)

Después de haber calculado las inecuaciones tanto de los obstáculos como las de la caja, el siguiente paso es determinar si existe una intersección entre el objeto y alguno de los obstáculos del medio ambiente. En el actual contexto, una intersección es equivalente a que la caja ha chocado con algún obstáculo. La intersección entre dos polígonos puede determinarse si es posible encontrar un segmento de línea recta que satisfaga el conjunto de restricciones de ambos polígonos. En particular, como se muestra en la Figura 3.6, existe colisión si un segmento de alguna arista del objeto A se encuentra dentro de B. El concepto general para realizar esta evaluación se describe a continuación.

3.2.1 Intersección entre una línea y un polígono

Primeramente, describiremos cómo se encuentra la intersección de una línea recta dada con un conjunto de restricciones lineales. Dada una línea definida por un punto x_0 y un vector director \vec{m}_0 , se localizan las intersecciones entre la línea y el conjunto de

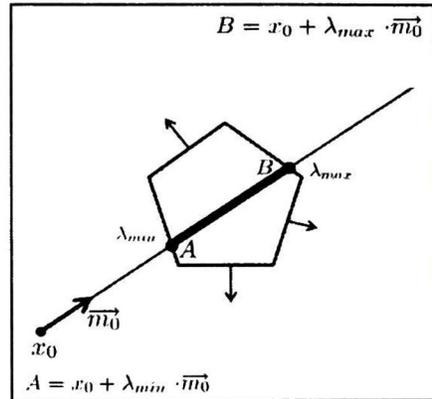


Figura 3.5: Obtención de un segmento de línea que satisface las restricciones de un polígono.

restricciones. Si las intersecciones de la línea generan un límite de intersección superior y otro inferior válido (la superioridad e inferioridad de la intersección se define por el sentido de la restricción que se intersecta con la línea), se habrá encontrado una colisión. (Figura 3.6)

Partiendo de la definición de la línea recta:

$$\vec{x} = \vec{x}_0 + \lambda \vec{m}_0, \lambda \in \mathbb{R}$$

y de la restricción i (del conjunto que define al polígono):

$$\vec{n}_i \cdot \vec{x} = d_i$$

es posible determinar el valor de $\lambda = \lambda_r$ para el punto de intersección. Obteniendo λ de estas ecuaciones:

$$\begin{aligned}\vec{n} \cdot (\vec{x}_0 + \lambda \vec{m}_0) &= d \\ \vec{n} \cdot \vec{x}_0 + \vec{n} \cdot \lambda \vec{m}_0 &= d \\ \vec{n} \cdot \lambda \vec{m}_0 &= d - \vec{n} \cdot \vec{x}_0 \\ \lambda(\vec{n} \cdot \vec{m}_0) &= d - \vec{n} \cdot \vec{x}_0 \\ \lambda_r &= \frac{d - \vec{n} \cdot \vec{x}_0}{\vec{n} \cdot \vec{m}_0} \text{ donde } \vec{n} \cdot \vec{m}_0 \neq 0\end{aligned}$$

La intersección entre la recta y la restricción lineal es una semirecta, que va de $\lambda = (-\infty, \lambda_r]$ o de $\lambda = [\lambda_r, +\infty)$, en función del signo de $\vec{n}_i \cdot \vec{m}_0$. En el primer caso se trata de un posible valor máximo λ^+ para el segmento que intersecta el conjunto de restricciones del polígono, mientras que en el segundo caso se trata de un posible valor mínimo λ^- . (Figura 3.5)

Finalmente, la intersección entre la recta y el conjunto de restricciones existe si $\min(\lambda^+) < \max(\lambda^-)$, y queda definida para el conjunto de valores $\lambda = [\max(\lambda^-), \min(\lambda^+)]$.

Este procedimiento se utiliza para determinar si existe una intersección entre dos polígonos convexos A y B. Para ello, se procede de la manera siguiente: a partir de la descripción semialgebraica de ambos polígonos

- $n_{A_i}x \leq d_{A_i}$
- $n_{B_i}x \leq d_{B_i}$

verificar para cada restricción $n_i x \leq d_i$ del polígono A si existe un segmento sobre la línea recta $n_i x = d_i$ que satisfaga el par de conjuntos de restricciones simultáneamente.

- Obtener, por rotación del vector normal n_i de la restricción de A, el vector director de la arista.

$$\vec{m}_0 = R \cdot \vec{n}_0 \text{ donde } R \text{ es una matriz de rotación de } 90^\circ.$$

- Calcular un punto cualquiera \vec{x}_0 sobre la línea. En nuestro caso, se toma el punto que se encuentra en la intersección de la línea y el vector perpendicular que une el origen con el vector \vec{m}_0 .

$$\vec{x}_0 = (\vec{d}_0 \cdot \vec{n}_{0x}, \vec{d}_0 \cdot \vec{n}_{0y})$$

- Se prueba la recta con todas y cada una de las restricciones de ambos polígonos.
 - Evaluar el resultado de $(\vec{m}_0 \cdot \vec{n}_j)$, si el resultado obtenido es igual a cero significa que no existe una intersección, por lo tanto \vec{m}_0 y \vec{n}_j son líneas paralelas.
 - Si $\vec{x}_0 \cdot \vec{n}_i \leq d_i$ significa que a pesar de ser líneas paralelas, \vec{x}_0 satisface la restricción, es por eso que la línea \vec{m}_0 será evaluada con el resto de las restricciones.
 - En otro caso, si las líneas son paralelas y \vec{x}_0 no satisface la restricción de distancia de \vec{n}_j , ya no será necesario continuar el cómputo, ya que se garantiza que no existe una intersección entre ellas..
 - El siguiente paso es calcular la distancia entre el punto \vec{x}_0 y la intersección con la restricción (λ).
 - Una vez calculado el valor de λ es necesario saber si este valor corresponde a un limite superior o a un limite inferior. Si $(\vec{n}_i \cdot \vec{m}_0) \leq 0$ significa que se trata de un limite inferior, en cualquier otro caso se tratará de un limite superior.

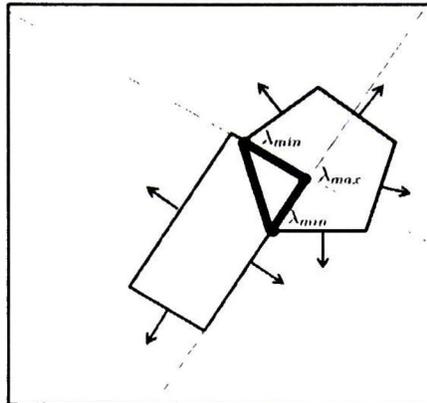


Figura 3.6: Ejemplo de 3 segmentos de línea que satisfacen el conjunto de restricciones de ambos polígonos.

- Dependiendo del valor obtenido de λ el siguiente paso sería actualizar el valor de los límites.
 - Si $\lambda_i > \lambda_{min} \Rightarrow \lambda_{min} = \lambda_i$
 - Si $\lambda_i < \lambda_{max} \Rightarrow \lambda_{max} = \lambda_i$
- El último paso es comparar los valores λ obtenidos; Si λ_{max} es mayor λ_{min} , se puede concluir que existe una intersección, en cualquier otro caso, se concluiría que la intersección no existe.
- Si existe intersección, el cómputo termina. Si no, entonces se prueba la siguiente arista de A.

En el algoritmo 3 se muestra la forma en la que se verifican las intersecciones entre dos polígonos. En la Figura 3.6 se muestra un ejemplo de la intersección entre dos polígonos. Como se puede ver, existen tres segmentos de línea que satisfacen el par de conjuntos de restricciones simultáneamente.

Algoritmo 3 Calcula colisiones

Entrada: Conjuntos de restricciones de dos polígonos
Salida: Identificador de colisión
ExiteColision ← Falso
 $\lambda_{max} \leftarrow \infty$
 $\lambda_{min} \leftarrow -\infty$
 $nRestric \leftarrow$ número de restricciones de objeto a y b
 $Restric[nRestric][3] \leftarrow$ Restricciones de objeto a y b $\{[x][y][d]\}$
para $i = 1$ hasta $i = nRestric$ **hacer**
 $m_{x_i}, m_{y_i} \leftarrow Restric[i][1] \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \{ \text{Obtener } \vec{m}_i \}$
 $x_i \leftarrow Restric[i][0] * Restric[i][2] \{ \text{Obtener componente } x \text{ de } \vec{x}_i \}$
 $y_i \leftarrow Restric[i][1] * Restric[i][2] \{ \text{Obtener componente } y \text{ de } \vec{x}_i \}$
 $\lambda_{temp} \leftarrow 0$
 bandera ← falso
 para $j = 1$ hasta $j = nRestric$ **hacer**
 si $i \neq j$ **entonces**
 si $(Restric[j][0] * m_{x_i} + Restric[j][1] * m_{y_i}) = 0$ **entonces**
 si $(Restric[j][0] * x_i + Restric[j][1] * y_i) \leq Restric[j][2]$ **entonces**
 bandera ← true
 sino
 Romper ciclo, no existe colisión
 fin si
 fin si
 si *bandera* \neq true **entonces**
 $\lambda_{temp} \leftarrow \frac{Restric[j][2] - ((Restric[j][0] * x_i) + (Restric[j][1] * y_i))}{(Restric[j][0] * m_{x_i}) + (Restric[j][1] * m_{y_i})}$
 si $\lambda_{temp} > \lambda_{min}$ **entonces**
 $\lambda_{min} \leftarrow \lambda_{temp}$
 sino si $\lambda_{temp} < \lambda_{max}$ **entonces**
 $\lambda_{max} \leftarrow \lambda_{temp}$
 fin si
 fin si
 fin para
 si $\lambda_{min} < \lambda_{max}$ **entonces**
 ExiteColision ← True
 Romper ciclo
 fin si
fin para
regresa *ExiteColision*

En la siguiente subsección se muestra la manera en la que utilizando los métodos previamente descritos, se construye una representación discreta del espacio de configuraciones, que nos será de utilidad para realizar el cálculo de trayectorias.

3.3 Espacio de configuraciones “C-space”

Teniendo la representación de todos los objetos involucrados en forma de inequaciones y con la capacidad de identificar si existe una intersección entre dos polígonos, el siguiente paso es construir formalmente el espacio de configuraciones (C-space). Este concepto, que es esencialmente una herramienta de representación, fue introducido desde la Mecánica en la Robótica por Lozano-Pérez [16]. El espacio de configuraciones es el conjunto de todas las configuraciones que puede tener el objeto (caja) en nuestro espacio. Dado que la caja puede desplazarse sobre el plano y girar sobre sí misma (como se muestra en la Figura 3.7), para definir su configuración es necesario fijar la posición de uno de los puntos de la caja y la orientación de esta respecto a un sistema de referencia fijo. La configuración de la caja vendrá dada por una tupla de dimensión tres (x, y, θ) que representa los grados de libertad de la caja, y el espacio de las configuraciones estaría formado por todas las posibles configuraciones, donde x e y vienen determinados por las dimensiones del espacio y θ tomará valores en el intervalo $[0, 2\pi]$.

La gran ventaja de realizar la planificación de movimientos en el espacio de configuraciones y no en el espacio real, es que en el espacio de trabajo real es necesario planificar los movimientos de todos y cada uno de los puntos del objeto, mientras que en el C-space solamente es necesario considerar la planificación de un único punto, por lo que es una tarea mucho más sencilla.

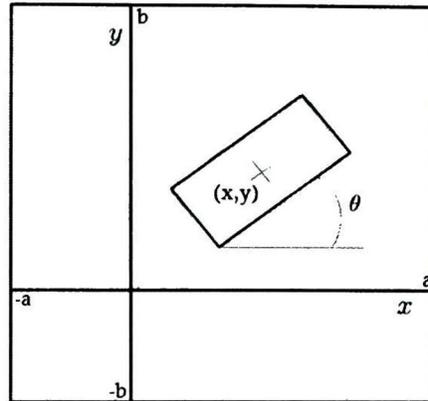


Figura 3.7: Configuración del objeto.

Por la dificultad de definir de manera precisa los límites del espacio de configuraciones libres, se utiliza una representación discretizada del mismo. La precisión de espacio de configuraciones está relacionada con la cantidad de configuraciones que se representen en el área de trabajo (resolución de la discretización), sin embargo, una mayor precisión no solamente afectará los tiempos de cálculo, si no que también afectará las necesidades de almacenamiento.

3.3.1 Expansión virtual de objetos

Con el objetivo de garantizar que el objeto nunca ocupe una posición pegada contra un obstáculo y que por la misma razón los robots no puedan manipular correctamente el objeto o ni siquiera poder rodearlo para ocupar los lugares adecuados de empuje, se aplicó un método de expansión virtual de los objetos. Es decir, en lugar de calcular el espacio de configuraciones del objeto a desplazar, se aumentan artificialmente sus dimensiones para incluir una distancia de seguridad mínima que deberá mantener el objeto con los obstáculos del medio ambiente. En la Figura 3.8 se muestra un ejemplo de

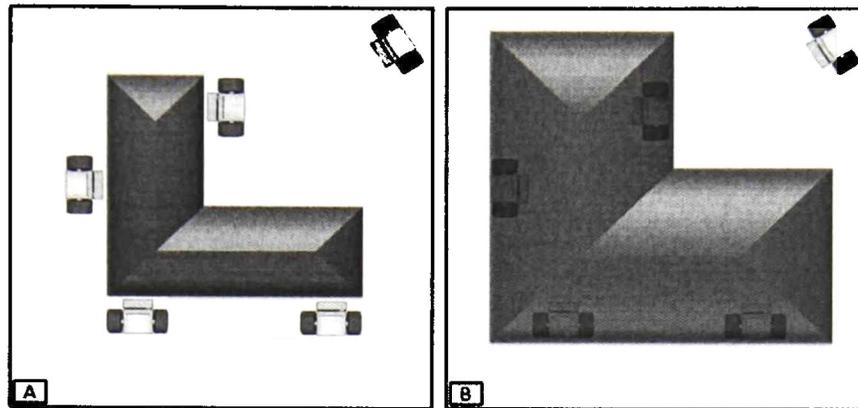


Figura 3.8: Ejemplo de la expansión virtual de un objeto.

la expansión de un objeto antes de iniciar el cálculo del espacio de configuraciones, como se puede ver el objeto crece de tal manera que engloba a los robots que lo rodean.

Este método para realizar la expansión de los objetos funciona de la siguiente manera.

- Definir una distancia mínima entre el objeto y los obstáculos. Esta distancia puede estar dada en función al tamaño de los robots que moverán el objeto.
- En base a la distancia previamente definida, incrementar las cotas del objeto.
- Realizar el cálculo del espacio de configuraciones con las nuevas dimensiones del objeto.
- Un vez que se tenga la ruta que se seguirá, hacer los cálculos restantes pero con las dimensiones originales del objeto.

Este mecanismo, además de solucionar los problemas antes descritos, nos permite obtener trayectorias que siempre mantienen una distancia adecuada entre el objeto y los obstáculos del ambiente. En la Figura 3.9 se muestra el ejemplo de una ruta

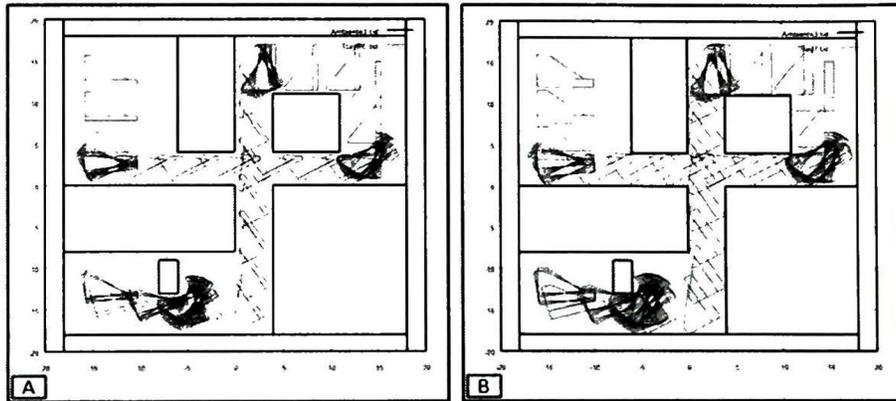


Figura 3.9: Ejemplo de 2 trayectorias idénticas, la Figura 3.9 A es representada con el tamaño original del objeto, 3.9 B es representada con el objeto expandido.

calculada con la expansión del objeto. La ruta fue calculada con el objeto extendido (ruta derecha) y la presentación de la ruta final (izquierda) muestra el tamaño original de objeto.

3.3.2 Discretización del espacio de configuraciones

En las siguientes líneas se explicarán detalladamente los pasos que se siguieron para realizar la discretización del C-space que se utilizó en el presente trabajo (Algoritmo 4).

- El primer paso en la construcción del C-space es la definición de la resolución de la matriz que lo representará. En esta implementación se manejan dos resoluciones distintas, una que define los pasos de rotación de la caja θ , directamente ligada con la cantidad de capas que tiene la matriz tridimensional, y el paso mínimo de desplazamiento horizontal y vertical del objeto a través del ambiente $[x, y]$. En el ejemplo mostrado en la Figura 3.10, se utilizó una resolución de 1 metro para movimientos horizontales y verticales y 5 grados de rotación en un ambiente de 40

x 40 metros.

- El siguiente paso es construir la matriz tridimensional de tamaño $[columnas, filas, capas]$, obtenidas al dividir las dimensiones del espacio de trabajo entre las resoluciones seleccionadas.
- El paso final es probar, para cada una de las configuraciones i representada en la matriz, existe colisión entre el objeto y uno o varios obstáculos del ambiente. Esto se realiza aplicando la transformación $A(i)$ para situar el objeto en el punto $[x_i, y_i, \theta_i]$, utilizando la función de cálculo de colisiones y de esta manera definir si el punto está libre de colisiones ($[x_i, y_i, \theta_i] = 0$) o si el lugar no es válido para la rotación y traslación definida, es decir, existe una colisión ($[x_i, y_i, \theta_i] = 1$).

Cabe hacer notar que la matriz de configuraciones es en realidad una matriz toroidal, ya que θ es un ángulo de rotación.

La Figura 3.10A se muestra la representación gráfica del C-space con una caja rotada 10 grados; las posiciones que colisionan no son mostradas en la Figura. En la Figura 3.10B se muestra la capas binaria del C-space correspondiente a la Figura 3.10A, los puntos marcados con 0 están libres de colisiones, mientras que los marcados con 1 son inválidos.

3.4 Resumen del capítulo

En este capítulo se mostraron las técnicas utilizadas para realizar las representaciones y transformaciones del ambiente y del objeto, estas manipulaciones nos permitieron en primera instancia obtener un mecanismo viable para detectar colisiones y así poder realizar el cálculo del espacio de configuraciones de un objeto. El espacio de configuraciones es el elemento primordial iniciar para el cálculo de trayectorias del objeto.

Algoritmo 4 Calcula C-space

Entrada: Conjuntos de restricciones del objeto A y de los obstáculos O
Salida: Espacio de configuraciones
 $hResol \leftarrow$ Resolución horizontal
 $vResol \leftarrow$ Resolución vertical
 $rotResol \leftarrow$ Resolución de rotación
 $capas \leftarrow \frac{360}{rotResol}$
 $columnas \leftarrow$ Tamaño horizontal ambiente / $hResol$
 $filas \leftarrow$ Tamaño vertical ambiente / $vResol$
 $Cspace[filas][columnas][capas] \leftarrow \emptyset$
para todos los obstáculos O de W **hacer**
 Calcula inecuaciones de O_x
fin para
para $i = 0$ hasta $i < capas$ **hacer**
 para $j = 0$ hasta $j < filas$ **hacer**
 para $k = 0$ hasta $k < columnas$ **hacer**
 Trasformación lineal $((k * hResol), (j * vResol))(i * rotResol)$
 Calcula inecuaciones de A
 si CalculaColisión = *true* **entonces**
 $Cspace[j][k][i] \leftarrow 1$
 sino
 $Cspace[j][k][i] \leftarrow 0$
 fin si
 fin para
 fin para
fin para
regresa $Cspace$

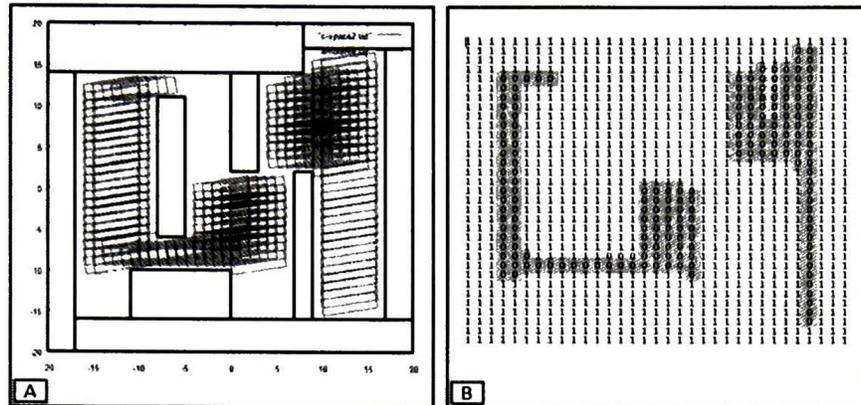


Figura 3.10: A) Representación gráfica del C-space con una caja rotada 10 grados. B) Representación binaria del C-space correspondiente a la Figura A.

En el siguiente capítulo se buscará construir una cadena de configuraciones válidas del objeto que unidas formen una ruta funcional para desplazar el objeto de un punto a otro. Un ejemplo de una cadena de configuraciones se muestra en la Figura 3.11.

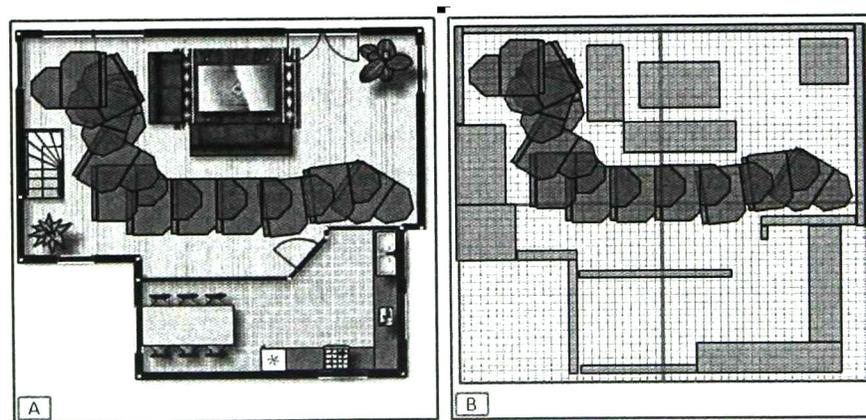


Figura 3.11: Cadena de configuraciones válidas que forman una ruta para desplazar un objeto de un punto a otro.

4

Cálculo de trayectorias

“Nos encontramos en los comienzos de una revolución robótica, en una situación parecida a la de la informática en los años 70.” Bill Gates [5].

Como se mencionó en los capítulos anteriores, en la actualidad se espera que los robots móviles realicen múltiples tareas en fábricas, almacenes, construcciones, ambientes peligrosos, etc. Para que estas expectativas se puedan convertir en una realidad los robots deberán estar preparados para trabajar en ambientes complejos. Por ello, un buen método de planeación de movimientos de objetos es muy importante para lograr completar las tareas eficientemente.

Autores como Laumond [24] han trabajado con el problema de planeación de movimientos para robots móviles, pero muy pocos se han enfocado a resolver el cálculo de trayectorias para objetos que serán desplazados por robots móviles no especializados,

dentro de un medio ambiente no estructurado.

En esta tesis se aborda el problema de planeación de trayectorias para un objeto que será desplazado por una comunidad de robots móviles, con la restricción de que sólo es posible empujar la caja. Para realizar esta tarea será necesario obtener en primera instancia la trayectoria de la caja, para después definir las trayectorias individuales que deberán seguir los robots para poder realizar la tarea.

En este capítulo se explicarán detalladamente la diferentes heurísticas que fueron confrontadas para solucionar el problema cálculo de trayectorias, incluyendo las heurísticas desarrolladas explícitamente para este trabajo; también se reportará el método utilizado para encontrar los puntos de empuje válidos, la construcción y características del algoritmo del frente de onda y algunos resultados preliminares.

4.1 Problema del cálculo de trayectorias

El problema del cálculo de trayectorias se puede definir como la búsqueda de una sucesión continua de posiciones de un robot o un objeto que permita llevarlo desde una posición inicial a una posición final. De acuerdo a los fenómenos físicos considerados, los planificadores de trayectorias pueden ser puramente geométricos o incluso dinámicos.

Los algoritmos de cálculo de trayectorias son responsables de mantener muchos de los aspectos esenciales del comportamiento de los agentes, incluyendo la evasión de colisiones y la ejecución de la tarea de forma precisa. El cálculo de trayectorias consume un tiempo de cómputo significativo en la mayoría de las simulaciones, particularmente en ambientes altamente dinámicos en donde los agentes se mueven simultáneamente.

Las características de la trayectoria obtenida estarán relacionadas a la distribución de los obstáculos a lo largo del ambiente de trabajo y por supuesto de la geometría del objeto que se desee desplazar. De esta manera, la topología del ambiente de trabajo restringirá el espacio libre de obstáculos por el cual se podrán construir las posibles trayectorias para alcanzar la posición final deseada.

Los algoritmos del planeación de trayectorias se pueden agrupar en dos categorías: enfoques locales y enfoques globales.

- Los algoritmos de planificación local se enfocan a la navegación en distancias cortas, donde el principal objetivo es la evasión de obstáculos. La navegación local permite que los robots se muevan en su ambiente inmediato y alcancen metas a corta distancia sin chocar con obstáculos. Es un enfoque que múltiples autores han abordado, como por ejemplo los trabajos de Ramírez [18] y de Sedighi [21]. Por desgracia, estos métodos al sólo tratar con información parcial del entorno que los rodea, pueden quedar “atascados” en mínimos locales¹.
- Por otro lado, la planificación global aborda el problema de la navegación en una escala más grande, en la cual, los robots pueden no observar directamente el punto destino, pero conocen totalmente el ambiente que los rodea, lo que les permite determinar, antes de desplazar el robot, si el problema de navegación admite una solución y calcular una ruta óptima. En la mayoría de los casos este tipo de planeación utiliza un mapa global para determinar su localización en el ambiente.

En este trabajo, nos basaremos en un método de planeación global, por lo que uno

¹Una función $f(x)$ alcanza un mínimo local en un punto a de su dominio si existe un entorno de a en el que los valores de f son mayores que $f(a)$.

de los requerimientos para construir el método propuesto es contar con un mapa del ambiente en el que se realizará la planeación de las trayectorias. Una vez obtenida la trayectoria de la caja, se podrán planificar los movimientos concretos de los robots y controlarlos para mantener al objeto en la trayectoria calculada.

4.2 Problema de la planeación de movimientos

Como se mencionó anteriormente, uno de los problemas fundamentales en la robótica móvil es la planeación de movimientos. De una forma básica este problema puede ser expresado de la siguiente forma: ¿De qué manera un robot puede decidir cuáles movimientos realizar para alcanzar la posición física deseada? Esta capacidad es muy necesaria en un robot móvil, si se parte de la definición en la que un robot es concebido como un ente que interactúa y se mueve en el mundo real. Debido a esto, lo mínimo que se puede esperar de un robot móvil es que sea capaz planificar sus propios movimientos.

4.2.1 Características de la planeación

A continuación se presentan una serie de conceptos necesarios para abordar el problema de la planeación de movimientos.

- **ESTADO:** Los problemas de planeamiento implican un “espacio de estados” que almacene todas las posibles situaciones que podrían presentarse a lo largo de la realización de la tarea. Por ejemplo, en nuestro caso, el “espacio de estados” que utilizaremos nos permitirá representar todas las posiciones y orientaciones que nuestro objeto puede tomar en el ambiente dado. Los “espacios de estados” pueden ser discretos (finitos y/o contables) y continuos (infinitos o incontables). En la mayoría de los casos, el tamaño del “espacio de estado” (en términos de

número de estados o de complejidad combinatoria) es demasiado grande para ser explícitamente representado, lo que nos obliga a realizar ciertas discretizaciones. En general, la definición de un "espacio de estados" es un componente importante en la formulación de un problema del planeación y en el diseño y el análisis de los algoritmos que pretendan resolverlo.

- **TIEMPO:** Todos los problemas de planeamiento implican una secuencia de decisiones que deben realizarse en un cierto plazo.
- **ACCIONES:** Un plan define las acciones que manipulan el estado. Los términos "acciones" y "operadores" son comunes en las ramas de la inteligencia artificial, pero, en teoría de control y robótica, los términos relacionados son "entradas" y "controles". En la formulación del planeamiento se debe especificar cómo cambian los estados cuando las acciones son aplicadas. Para la mayoría de los problemas de planeación de movimientos, la referencia explícita al tiempo es evitada especificando una trayectoria que cruce de manera continua un espacio de estados. Para nuestro proyecto, las trayectorias pueden ser especificadas como los cambios de posición del objeto cuando las acciones de empuje son aplicadas.
- **ESTADO INICIAL Y FINAL:** Un problema del planeamiento implica generalmente el comenzar en algún estado inicial para tratar de alcanzar un estado final específico. Es claro que las acciones a realizar intentarán acercarnos al estado final.
- **CRITERIO DE EVALUACIÓN:** Será la manera en que se calificará el plan obtenido. En general existen dos tendencias principales para definir un criterio de evaluación:
 1. Viable: Encontrar un plan que nos lleve a un estado final, sin importar su eficiencia.

2. Óptimo: Encontrar un plan que además de ser viable y de guiarnos a un estado de final optimice algún criterio de desempeño.
- UN PLAN: Un plan generalmente impone una estrategia o un comportamiento específico ante una toma de decisión. Un plan puede especificar simplemente una secuencia de acciones que deberán de ser ejecutadas.

En las siguientes subsecciones, se describirán los algoritmos y heurísticas que se utilizaron para construir la estrategia propuesta en esta tesis.

4.3 Algoritmo de frente de onda

Como se mencionó con anterioridad, el algoritmo de frente de onda nos permite calcular la ruta más corta entre dos puntos de un ambiente con obstáculos. Se trata de un esquema inspirado de los campos de potencial artificial, con la ventaja de que no presenta mínimos locales. El único requerimiento previo para su construcción es el contar con una matriz de ocupación que represente los puntos que son válidos para transitar y aquellos que no lo son. En este caso, nosotros partiremos de la matriz del C-space que se construyó en los pasos previos.

El algoritmo del frente de onda es ampliamente usado para el cálculo de trayectoria de robots móviles. Las características de este método lo hacen ser uno de los más utilizados actualmente en la robótica móvil. Entre sus características más destacables se encuentran las siguientes:

1. Este método nos permite calcular trayectorias libres de colisiones (si estas existen).
2. A diferencia de métodos como los basados en campos de potenciales, el algoritmo de frente de onda no tiene mínimos locales.

3. Este método nos garantiza encontrar la ruta más corta entre el origen y el destino.

La construcción de un frente de onda según LaValle [12] se realiza de la siguiente manera. Supóngase que el costo mínimo común entre cada estado es i , y se tiene un conjunto de estados organizados en un frente de onda, W_i . El punto de inicio del frente de onda es W_0 y representa el estado objetivo X_G . El algoritmo asignará el costo de traslado óptimo de 1 para todos los estados que puedan ser alcanzados desde W_0 en un solo paso. Los estados que reciban costo de 1 pueden ser organizados en el frente de onda como W_1 . Los vecinos inexplorados de W_1 tendrán un costo asignado de 2. Este proceso se repetirá desde i hasta $i + 1$ hasta que todos los estados alcanzables hayan sido visitados. Todo estado no alcanzable se le asignará el valor de $\phi(x) = \infty$. El costo óptimo de moverse de un estado a otro será calculado en tiempo $O(n)$, en donde n es el número de estados alcanzables.

Partiendo de lo anterior el robot debe seguir el gradiente descendente seleccionando el estado vecino con el menor costo, repitiendo el proceso hasta alcanzar el estado objetivo X_G . El criterio para la selección del vecino se muestra a continuación.

$$u^* = \operatorname{argmin}_{u \in U(x)} \{ \phi(f(x, u)) \}$$

En donde $U(x)$ es el conjunto de espacios de acción vecinos al estado actual x hacia los cuales se puede desplazar el robot. u^* representa el siguiente estado y $f(x, u)$ es la función de transición de estado para la acción u aplicada al estado x .

4.3.1 Construcción del frente de onda

Partiendo de una matriz de ocupación, en la que el espacio de estados es discretizado como un conjunto de puntos uniformemente distribuidos, y en la que los estados libres

son identificados con ceros y los estados no válidos con valores negativos, el algoritmo de propagación funciona básicamente de la siguiente forma:

- El primer paso es marcar el punto destino con el valor cero. Entonces el algoritmo de frente de onda marcará cada celda adyacente a la meta con un valor equivalente al costo de desplazarse desde ellas hacia la meta.
- El siguiente paso es repetir de manera recursiva el marcado de los vecino adyacentes de las celdas ya marcadas.
- El algoritmo concluirá con el marcado de las casillas hasta que todas las celdas alcanzables desde la meta estén marcadas con un valor equivalente su distancia a la meta.

Como en nuestro caso, la matriz de ocupación es tridimensional (un estado de la caja es la tripleta (x, y, θ)), es necesario incluir la validación de las casillas vecinas de las capas superior e inferior de la celda considerada.

La propagación del frente de onda es una versión especializada del algoritmo de Dijkstra, que optimiza el número de transiciones de estados para alcanzar el objetivo, sin la necesidad de construir y mantener una cola de prioridades (Algoritmo 5).

La Figura 4.1 muestra un ejemplo de la aplicación del algoritmo de frente de onda a una matriz bidimensional, donde la casilla D representa el estado destino.

El valor de la casilla origen, representa la distancia a la que se encuentra de la casilla final. Dado lo anterior, la construcción de trayectoria se realiza siguiendo las casillas marcadas de forma decremental, partiendo de la casilla inicial. En la Figura 4.2 se muestra un ejemplo de una trayectoria que se construyó a partir de la matriz marcada con el

Algoritmo 5 Propagación de frente del onda**Entrada:** Matriz de espacio de configuraciones y nodo destino**Salida:** Matriz de frente de onda $queue \leftarrow \emptyset$ $n_f \leftarrow$ Nodo destino $valn \leftarrow 0$ {Valor del destino}Insertar n_f a la cola $queue$ **mientras** $queue \neq \emptyset$ **hacer** $n \leftarrow$ primer nodo de la cola $queue$ $valn \leftarrow$ valor del primer nodo de la cola $queue$ **para** Todos los vecinos de n **hacer** $n_v \leftarrow$ vecino de n $valn_v \leftarrow$ valor de vecino de n **si** n_v es una configuración válida y n_v no está en la cola $queue$ y $valn_v$ no ha sido actualizado **entonces** Insertar n_v a la cola $queue$ Valor de vecino de $valn_v \leftarrow valn + 1$ **fin si** **fin para** Remove el primer elemento de la cola $queue$ **fin mientras**

| | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|
| 18 | 17 | 16 | 15 | 14 | 13 | 12 | 13 | 12 | 11 | 10 |
| 17 | 16 | 15 | 14 | 13 | 12 | 11 | 12 | 11 | 10 | 9 |
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 10 | 9 | 8 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | | 9 | 8 | 7 |
| 14 | | | | 10 | 9 | 8 | | | | 6 |
| 13 | 12 | 11 | | 9 | 8 | 7 | 6 | 5 | | 5 |
| 12 | 11 | 10 | | 8 | 7 | 6 | 5 | 4 | | 4 |
| 11 | 10 | 9 | | | | 5 | 4 | 3 | 2 | 3 |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | D | 1 |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 |

Figura 4.1: Ejemplo de una matriz de marcada con el algoritmo de frente de onda.

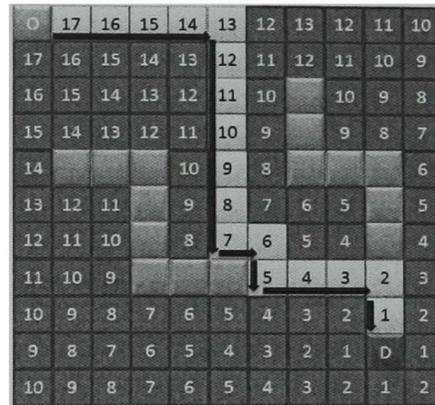


Figura 4.2: Ejemplo de una ruta construida a partir del C-space marcado con el algoritmo del frente de onda.

algoritmo del frente de onda. Como se puede ver, además de la trayectoria mostrada, es posible construir una gran cantidad de rutas distintas entre el origen y el destino siguiendo el método antes descrito, dado que una celda generalmente tiene más de un vecino con valor mínimo. Entonces, el objetivo será encontrar aquella ruta que sea más eficiente según algún criterio.

4.4 Modelo de navegación

El modelo de marcado para las casillas vecinas que será utilizado en este proyecto es el conocido como Manhattan, que obtiene su nombre debido a que en \mathbb{R}^2 corresponde a la longitud de una trayectoria moviéndose en una rejilla de ejes alineados. Por ejemplo, la distancia entre $(0,0)$ y $(2,5)$ es de 7 unidades utilizando este método, ya que son necesarios 2 movimientos hacia el este y 5 hacia el norte, lo que sucedería de manera real al movernos en las calles de alguna ciudad. Por lo tanto el modelo Manhattan nos permitirá realizar movimientos contenidos en un conjunto de acción U de vectores en donde sólo un componente es distinto a cero y podrá tomar los valores de 1 o -1. Por

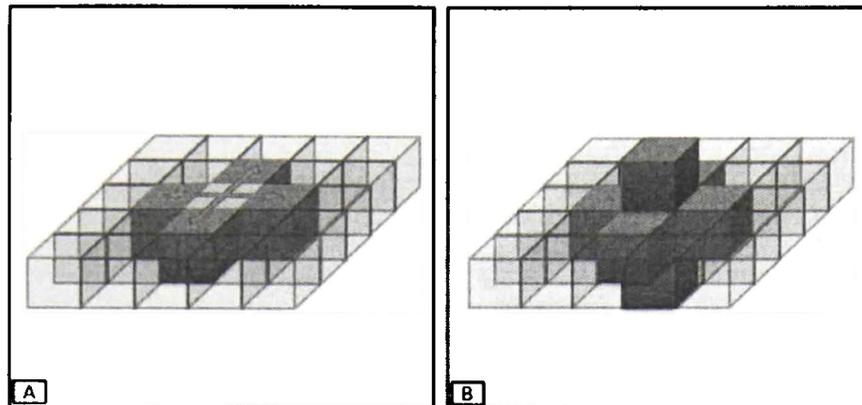


Figura 4.3: Ejemplo de una vecindad de 4 y 6 elementos.

ejemplo, para \mathbb{R}^2 , el conjunto de acciones (desplazamientos) posibles es:

$$U = \{(1, 0), (-1, 0), (0, -1), (0, 1)\}$$

Este grupo de acciones produce el conjunto de movimientos "norte", "sur", "este" y "oeste". Sin embargo, como ya se había mencionado, en este proyecto la matriz de ocupación es tridimensional, por lo que se realiza una evaluación basada en una vecindad de 6, y el conjunto de acciones contiene también los desplazamientos "arriba" y "abajo" (Figura 4.3).

4.4.1 Reconfiguraciones

Al trabajar con robots móviles no especializados en manipulación de objetos es necesario, en algunos casos, modificar la disposición de estos alrededor de la caja (reconfiguración) para poder obtener el empuje adecuado y así poder desplazar el objeto en la dirección deseada.

Como se mencionó en capítulo 1, el objetivo de esta tesis es desarrollar una estrategia que no sólo permita construir rutas cortas entre la posición inicial de la caja y su posición final, sino poder determinar aquellas rutas que requieran el menor número de cambios de dirección de la caja, para que los cambios de posiciones de los robots con respecto a la caja también sean en número reducido. Para lograr esto, se proponen tres heurísticas que se enfocan en la correcta selección de los vecinos de las celdas durante la construcción de la ruta. Esta selección estará enfocada a la reducción de cambios de dirección del objeto.

La reducción de reconfiguraciones de los robots a lo largo de la trayectoria permite, en primera instancia, garantizar una importante reducción de tiempo debido a que el simple cambio de puntos de empuje de los robots es un proceso que lleva cierto tiempo en lograrse. Otro punto que se puede mejorar al reducir las reconfiguraciones de los robots es el incremento de la precisión de los movimientos, ya que los continuos movimientos de los robots alrededor de la caja pueden aumentar el riesgo de colisiones no deseadas entre ellos y por supuesto, el incremento del error en la odometría por el constante movimiento de los robots.

En la Figura 4.4 se muestra un ejemplo de una ruta construida sin ningún tipo de heurística especializada para la selección entre vecinos con igual valor mínimo. Cada cambio de dirección del objeto requiere que los robots modifiquen su posición para poder seguir la ruta planeada.

4.4.2 Selección de vecinos

En la sección anterior se mencionó que, en función de la selección de vecinos con igual valor mínimo, se pueden obtener rutas muy diferentes entre la celda origen y la celda destino. Con el fin de evaluar las capacidades de la estrategia propuesta en este trabajo

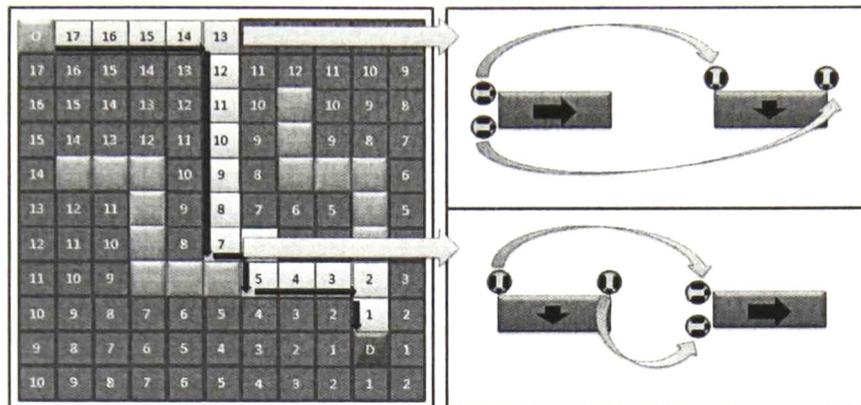


Figura 4.4: Ejemplo de una ruta con múltiples reconfiguraciones

de investigación, se implementaron dos mecanismos simples de selección de vecinos para realizar la construcción de las rutas. Estos métodos de selección de vecinos sólo buscan conectar el origen y el destino sin ninguna evaluación suplementaria. Los mecanismos implementados son los siguientes:

1. Selección de vecino aleatoria.- En el caso de que exista más de un vecino válido, la selección de la siguiente celda se realizará de manera totalmente aleatoria.
2. Selección de vecino con orden predefinido.- La ruta es obtenida a partir de una selección del vecino válido en la existe un orden específico para la evaluación de las casillas, por ejemplo: Norte, Sur, Este, Oeste, Arriba y Abajo, en ese orden. Así, si Sur y Arriba son los vecinos de valor mínimo, se elegirá Sur como acción siguiente.

En la Figura 4.5 se muestran dos trayectorias generadas con los métodos antes descritos. La ruta A se obtuvo con el método de selección de vecino aleatorio registrando 65 cambios de dirección y la ruta B con la de selección de vecino con orden predefinido con 36 cambios de dirección.

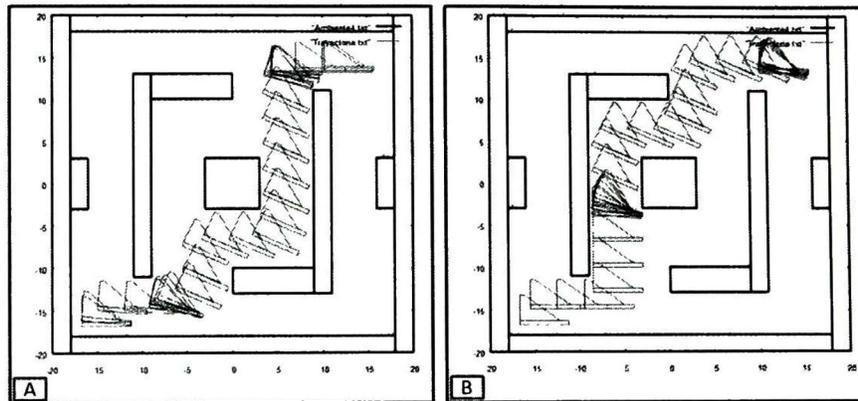


Figura 4.5: Rutas calculadas con las heurísticas de selección de vecino aleatorio A y evaluación en orden predefinido B

Las heurísticas propuestas para calcular rutas con un menor número de cambios de dirección se obtienen por medio de una selección de vecinos especializada. Las heurísticas propuestas se listan a continuación:

1. "Medición de alcance"
2. "Medición del alcance con validación de capas"
3. "Construcción de grafo de ruta"

4.4.3 Heurística de "Medición del alcance"

La heurística de "Medición de alcance" para la selección de la casilla vecina trata de reducir la cantidad de reconfiguraciones de los robots a lo largo de la trayectoria. Se trata de un algoritmo voraz, aplicado de la siguiente manera:

Partiendo de la matriz de frente de onda e iniciando desde un nodo origen válido n_i , el primer paso es evaluar todos los vecinos de n_i que son alcanzables en un solo movimiento

(norte, sur, este, oeste, superior e inferior), si existen al menos 2 nodos vecinos con un valor inferior al de n_i la selección dependerá del que tenga la mayor "medición de alcance".

La "Medición de alcance" es equivalente a conocer el número de casillas que podemos recorrer de manera decreciente al tomar una dirección determinada. Se realiza siguiendo los nodos vecinos válidos con valores decrecientes, sin cambiar de dirección y finaliza al encontrar el primer nodo que termine la cadena de decrementos o que sea inválido.

Un ejemplo de esta estrategia es la siguiente: Si se parte del nodo n_i que tiene un valor de 99 y los únicos vecinos con valores menores al 99 son los de las casillas oeste con un valor de 98 y superior con un valor también de 98, el siguiente paso sería contabilizar las casillas con dirección oeste y superior hasta terminar el decremento de los valores de los vecinos de la misma dirección. La selección del nodo vecino que se utilizará quedará determinada al comparar las mediciones de alcance en ambas direcciones; evidentemente, la dirección que tenga un mayor alcance será la ganadora. El proceso de medición será repetido iterativamente hasta llegar al nodo destino n_f (Algoritmo 6).

En la Figura 4.6B se muestra un ejemplo de una trayectoria obtenida al aplicar esta heurística. Si se le compara con la trayectoria mostrada en la Figura 4.6A que utiliza el método de selección basada en orden predefinido, podemos comprobar que se reduce de manera importante el cambio de direcciones de la caja y por ende, el número de reconfiguraciones de los robots alrededor de la caja también es reducido, lo que se traduce directamente en rutas con mayor continuidad.

El funcionamiento básico de esta heurística es el siguiente:

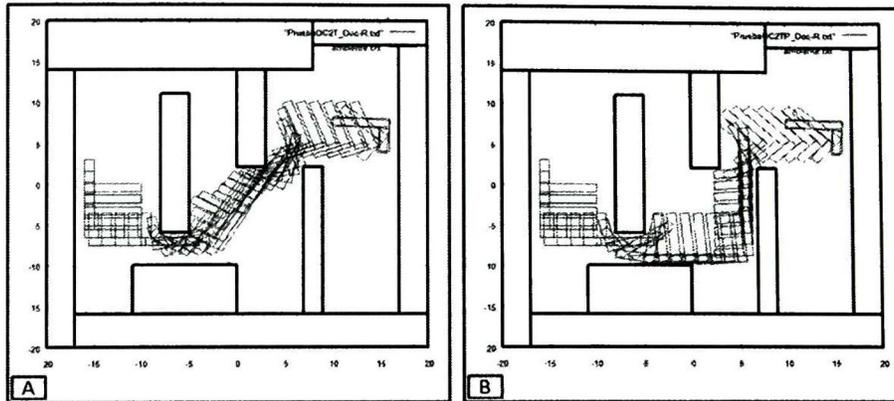


Figura 4.6: A) Representación gráfica de la trayectoria con la heurística de selección de vecinos predefinido. B) Trayectoria obtenida con la aplicación de la heurística propuesta para reducción de reconfiguraciones.

1. Iniciar desde un punto válido.
2. Validar todos los vecinos (norte, sur, este, oeste, superior e inferior).
3. Realizar las mediciones de alcance en todas las direcciones candidatas.
4. Seleccionar la dirección que tenga un mayor alcance.
5. Repetir los pasos 2-4 hasta que se alcance el destino.

4.4.4 Heurística de “Medición del alcance con validación de capas”

Una vez que se ha logrado obtener una ruta que garantiza tener mayor continuidad de movimientos, el siguiente aspecto a mejorar son los puntos de empuje de los robots.

Algoritmo 6 Heurística 1 "Medición de alcance"**Entrada:** Matriz de frente de onda y nodo origen n_i **Salida:** Trayectoria de n_i a n_f (si existe alguna) $nodo_x \leftarrow$ nodo origen n_i insertar nodo $nodo_x$ en *ruta***mientras** $nodo_x \neq$ nodo destino n_f **hacer** **para** todos los vecinos de $nodo_x$ **hacer** **si** vecino de $nodo_x$ es válido **entonces** guardar dirección válida en *direcV* **fin si** **fin para** **si** existen 2 o mas direcciones válidas **entonces** **para** todas las direcciones válidas *direcV* **hacer** *alcance* \leftarrow cantidad de nodos que tienen decrementos en sus valores
 siguiendo la dirección *direcV* y partiendo desde $nodo_x$ **fin para** **fin si** construir ruta siguiendo la dirección *direcV* con el valor máximo de *alcance* $nodo_x \leftarrow$ último nodo del segmento de ruta con dirección *direcV***fin mientras****regresa** *ruta*

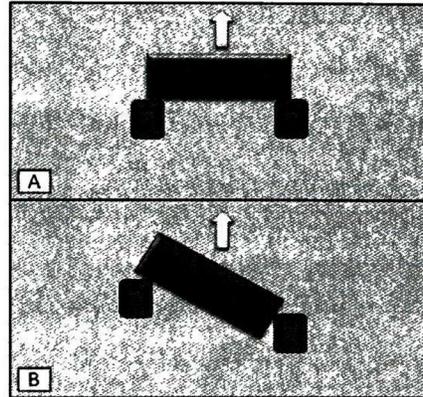


Figura 4.7: Ejemplo de 2 configuraciones distintas de la caja en donde los puntos de empuje son afectados por la rotación.

El trabajar con robots no especializados en tareas de manipulación de objetos limita las capacidades de los robots de tener contactos adecuados con el objeto. Los puntos de empuje son las áreas del objeto en las que los robots hacen contacto con él. Un incremento en el área de contacto entre el robot y el objeto reduciría la probabilidad de tener desplazamientos inadecuados lo que podría traducirse en una mayor precisión en las tareas de empuje.

Para mejorar las áreas de contacto entre los robots y el objeto, tomando en cuenta las direcciones de los movimientos implementados (norte, sur, etc.) proponemos privilegiar ciertas orientaciones de la caja (0° , 90° , 180° y 270°).

En la Figura 4.7 se muestra la comparación de dos configuraciones de los robots, con diferentes orientaciones de la caja y con la misma dirección de empuje.

Como se puede ver, es posible mejorar el área de contacto entre el objeto y el robot si se privilegian algunas rotaciones del objeto. Aplicar esta acción mejoraría los puntos de

empuje para los movimientos norte, sur, este y oeste cuando el objeto se encuentre en alguna de las rotaciones privilegiadas.

El modelado del espacio que se utiliza en este proyecto nos permite identificar las rotaciones del objeto por medio de las capas del espacio de configuraciones, por lo tanto es posible construir rutas que den privilegios a la utilización de ciertas capas del espacio de configuraciones.

Para poder calcular rutas que aumenten la continuidad de los movimientos y al mismo tiempo mejore los puntos de empuje proponemos utilizar la heurística de "Medición de alcance" pero con algunos criterios adicionales. La descripción de las modificaciones se muestra a continuación.

En la sección anterior se sugería que la medición alcance de las rutas se realizara al inicio del proceso y cada vez que se llegara al final de un segmento de ruta. Para poder privilegiar el uso de ciertas capas del C-space proponemos realizar mediciones de alcance no sólo al llegar al final de un segmento de ruta, si no también al que la ruta seleccionada atravieza alguna de las capas correspondientes a las orientaciones que se desean privilegiar. Esto implica que durante la rotación del objeto, se determine si el detener la rotación en alguna orientación específica permite obtener rutas de longitud equivalente, en el siguiente paso.

Estas acciones claramente arrojarían dos resultados: Uno, que a pesar de estar siguiendo una dirección especificada, se pueda medir nuevamente en búsqueda de nuevas direcciones y elevar así la probabilidad de encontrar nuevos segmentos de ruta más acordes a la geometría del objeto. Dos, el hecho de colocar nuevos puntos de medición

pueden elevar sensiblemente la cantidad de cambios de dirección del objeto y por lo tanto elevar también la cantidad de reconfiguraciones de los robots.

El funcionamiento básico de esta heurística es el siguiente:

1. Iniciar desde un punto válido.
2. Validar todos los vecinos (norte, sur, este, oeste, superior e inferior).
3. Realizar las mediciones de alcance en todas las direcciones válidas.
4. Seleccionar la dirección que tenga un mayor alcance.
5. Si al momento de construir el segmento de ruta se pasa por una casilla que pertenezca a una de las capas privilegiadas repetir desde esa casilla los pasos 2-4.
6. Repetir los pasos 2-5 hasta que se alcance el destino.

Adicionalmente, es posible inducir al esquema a tratar de alcanzar las capas privilegiadas, utilizando para ello un coeficiente de ponderación:

Si nos encontráramos en un nodo cualquiera n_x , con 3 posibles direcciones a seguir, norte, este y capa superior. Después de realizar la medición de alcance encontramos que la longitud de los segmentos de rutas son los siguientes: norte 21, este 19 y capa superior 11. Si la ruta con dirección a la capa superior pasa por algunas de las capas privilegiadas, duplicaríamos su valor con la intención de inducir un cambio de capa. Al momento de decidir la dirección que se deberá seguir, se tomaría la del segmento de ruta con dirección hacia la capa superior (ahora con un valor de 22). Realizar esta acción nos permitiría pasar por alguna de las capas privilegiadas y realizar una nueva medición de alcance en ella, lo

que en el mejor de los casos nos permitirá encontrar un nuevo segmento de ruta más largo.

Algoritmo 7 Heurística 1 "Medición de alcance con validación de capa"

Entrada: Matriz de frente de onda y nodo origen n_i

Salida: Trayectoria de n_i a n_f (si existe alguna)

$priv \leftarrow$ capas correspondientes a las rotaciones que se desean privilegiar

$nodo_x \leftarrow$ nodo origen n_i

insertar nodo $nodo_x$ en $ruta$

mientras $nodo_x \neq$ nodo destino n_f **hacer**

para todos los vecinos de $nodo_x$ **hacer**

si vecino de $nodo_x$ es válido **entonces**

 guardar dirección válida en $direcV$

fin si

fin para

si existen 2 o mas direcciones válidas **entonces**

para todas las direcciones válidas $direcV$ **hacer**

$alcance \leftarrow$ cantidad de nodos que tienen decrementos en sus valores siguiendo la dirección $direcV$ y partiendo desde $nodo_x$

si siguiendo la dirección $direcV$ se pasa por alguna capa $priv$ **entonces**

$alcance \leftarrow alcance * 2$

fin si

fin para

fin si

 construir ruta siguiendo la dirección $direcV$ con el valor máximo de $alcance$

si al construir ruta el $nodo$ actual pertenece a alguna de las capas $priv$ **entonces**

$nodo_x \leftarrow$ $nodo$ actual

sino

$nodo_x \leftarrow$ último nodo del segmento de ruta con dirección $direcV$

fin si

fin mientras

regresa $ruta$

En la Figura 4.8 podemos comparar las heurísticas de "Medición de alcance" y "Medición de alcance con validación de capa". Para este ejemplo la ruta con la heurística simple registro 10 cambios de dirección mientras que la que incluye la validación de capa registro 11, pero esta última mejorando los puntos de empuje de uno de los movimientos

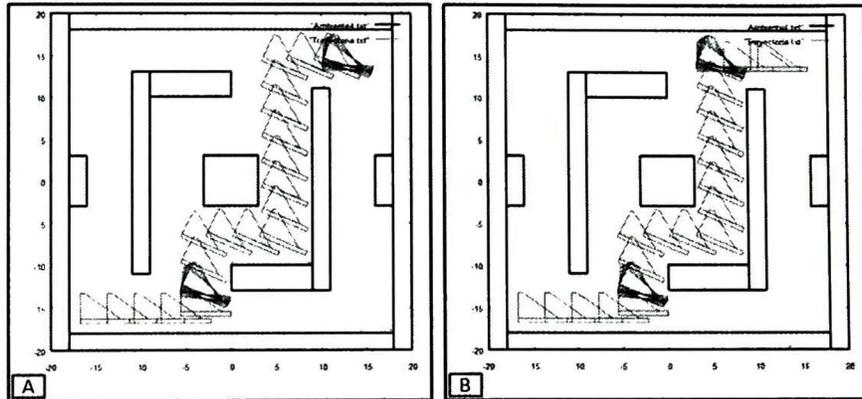


Figura 4.8: Ejemplo de rutas generadas con las heurísticas de “Medición de alcance” y “Medición de alcance con validación de capa”.

principales de la ruta.

4.4.5 Heurística de “Construcción de grafo ruta”

Con el objetivo de disminuir aún más la cantidad de cambios de dirección de las rutas, se propuso una tercera heurística basada en la construcción de un grafo en el que cada nodo representaría un cambio de dirección del objeto. Utilizar esta nueva representación nos permitirá construir la ruta basándonos en la rama más corta del grafo; esta acción estará directamente ligada a la reducción de cambios en la dirección del objeto y por lo tanto a la reducción del número de reconfiguraciones necesarias de los robots para desplazar el objeto.

La forma en la que el grafo es construido es la siguiente:

1. Iniciar desde un punto válido y agregarlo como nodo raíz.
2. Realizar mediciones de alcance todos los vecinos (norte, sur, este, oeste, superior e inferior).

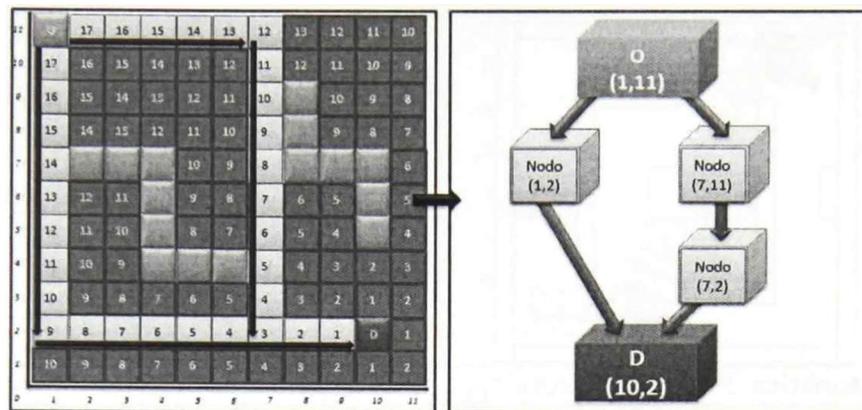


Figura 4.9: Ejemplo de construcción del árbol de ruta.

3. Todas las casillas al final de cada medición serán agregadas como nodos del grafo, unidos al nodo raíz.
4. Después de haber agregado todos los nodos para las direcciones válidas, repetir los pasos 2 y 3 pero ahora iniciando desde los nodos agregados.
5. La construcción del árbol será finalizada cuando todas las ramas converjan al punto destino.

En la Figura 4.9 se muestra un ejemplo de la construcción del árbol de ruta partiendo de un espacio de configuraciones marcado con el algoritmo del frente de onda.

Para mejorar los puntos empuje de esta estrategia se propone realizar un post-procesamiento especializado para la selección de puntos de empuje óptimos, con lo cual se separará el cálculo de rutas de la selección de puntos de empuje. En la sección de "Obtención de puntos de empuje" se describe detalladamente el proceso de detección de puntos de empuje válidos.

En la Figura 4.10 A se muestra una ruta de ejemplo construida con la heurística de "Grafo de ruta". Esta ruta registró, para este ejemplo, 10 cambios de dirección. En la

Algoritmo 8 Heurística 3 "Grafo de ruta"**Entrada:** Matriz de frente de onda y nodo origen n_i **Salida:** Trayectoria de n_i a n_f (si existe alguna) $posOrigen \leftarrow$ nodo origen n_i Insertar nodo $posOrigen$ en $nodo[0]$ $x \leftarrow 1$ **mientras** $nodo[x] \neq \emptyset$ **hacer** $posOrigen \leftarrow$ posición de $nodo[x]$ **para** Todos los vecinos de $nodo[x]$ **hacer** **si** Vecino de $nodo[x]$ es válido **entonces** $dir \leftarrow$ dirección para alcanzar el vecino de $nodo[x]$ $posVec \leftarrow$ posición del vecino de $nodo[x]$ **mientras** Vecino de $posVec$ en dirección dir sea válido **hacer** $posVec \leftarrow$ posición del vecino de $posVec$ **fin mientras** Insertar nodo $posVec$ en $nodo[]$ **fin si** **fin para** $x++$ **fin mientras****regresa** grafo de rutaConstruir ruta en base al nodo final de menor profundidad

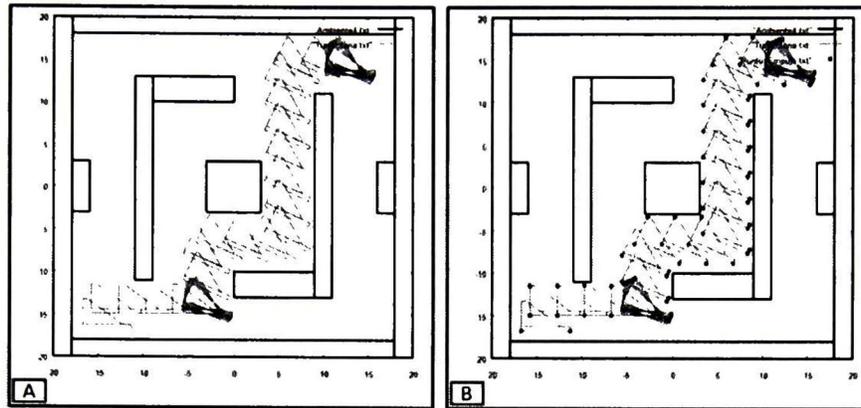


Figura 4.10: A) Ejemplo de ruta construida con la heurística de "Grafo de ruta". B) Puntos de empuje válidos para la ruta.

Figura 4.10 B se muestra la misma ruta pero en la que se grafican los puntos de empuje válidos para realizar la tarea.

4.5 Obtención de puntos de empuje

Cualquiera que sea la heurística aplicada, la ruta del objeto es entonces una secuencia de trayectorias dirigidas, que requieren una reconfiguración de los robots para pasar de una a la otra. Para cada segmento de la trayectoria, pueden existir diversas combinaciones de posiciones de los robots que pueden lograr el desplazamiento deseado. Con los puntos de empuje detectados, sería entonces posible aplicar algún mecanismo de optimización que permita distribuir los robots de manera eficiente alrededor del objeto, de manera que un mismo robot pueda participar en dos secciones consecutivas de la trayectoria sin tener que cambiar de posición. Es importante entonces contar con un algoritmo que permita detectar los puntos que son congruentes con la dirección que se desea dar al objeto, y descartar así los puntos de arrastre.

En este trabajo se considera que los puntos del objeto que pueden ser utilizados por los robots como puntos de empuje han sido definidos por el usuario. En un trabajo futuro, estos puntos podrían ser calculados automáticamente según las necesidades de la tarea. Para mostrar la validez del enfoque, se considera que los posibles puntos de empuje son todos los vértices externos de los polígonos que conforman el objeto. La forma en la que se calculan los puntos de empuje es la siguiente.

Para cada sub ruta:

- Obtener el vector que define el movimiento de la caja dv , por ejemplo: para la dirección norte el vector sería el $(0,1,0)$, para la dirección sur el vector sería el $(0,-1,0)$. En el caso de las rotaciones $(0,0,-1)$ si es en sentido horario y $(0,0,1)$ en sentido antihorario.
- Para cada punto de empuje definido para el objeto, obtener la proyección del vector dv en las dos restricciones que definen el vértice.
- Si al menos una de las dos proyecciones es negativa, el punto se considerará válido para empuje. Si por el contrario, ambos resultados son cero o positivos el punto será descartado para empuje (punto de arrastre).

En la Figura 4.10 B se resaltan los puntos que se consideran válidos para empuje en algunos puntos de la trayectoria obtenida.

4.6 Algoritmo A* (A-Estrella) para el cálculo de trayectorias

Para poder realizar comparaciones pertinentes de las heurísticas propuestas, se implementó un algoritmo A-Estrella (A*). Este algoritmo fue presentado en 1968 por

Hart [8] y entre sus múltiples aplicaciones, es utilizado para calcular trayectorias de robots móviles.

Existen numerosos algoritmos diseñados para buscar una trayectorias en grafos pesados, por ejemplo:

- Algoritmo de Dijkstra, que garantiza encontrar las trayectorias más cortas entre los vértices. El algoritmo de Dijkstra trabaja etiquetando los nodos de la red con números. Al principio, todos tienen la etiqueta 00 excepto el nodo inicial a que tiene la etiqueta 0. Los arcos tienen un peso w_{ij} que representa la distancia del enlace (i, j). El algoritmo de Dijkstra visita y reenumera los nodos, de manera que cuando el nodo z tiene una etiqueta permanente, se ha obtenido la solución final.
- BFS (Best-First-Search), que funciona de una manera algo similar al algoritmo de Dijkstra, salvo que BFS trabaja con ciertas heurísticas para realizar las estimaciones de distancia al objetivo. Esta característica le permite seleccionar el nodo más cercano al objetivo en vez seleccionar el nodo más cercano al origen (de acuerdo a la heurística que se emplee). No obstante sus ventajas, los algoritmos BFS no garantizan encontrar las trayectorias más cortas, sin embargo, funcionan mucho más rápido que el algoritmo de Dijkstra porque utilizan una heurística para dirigirse rápidamente hacia el objetivo.

El algoritmo A* combina las capacidades del algoritmo del Dijkstra y de los BFS, ya que busca las rutas más cortas de manera similar a como lo hace Dijkstra y además incluye una heurística como en los BFS puede dirigirse rápidamente al objetivo, por lo que en algunos casos puede ser tan rápido como un BFS.

El secreto del éxito del algoritmo A* es que combina fragmentos de información

que Dijkstra (favoreciendo la selección de nodos cercanos al origen) con información que los BFS utilizan (que favorecen los nodos que están cercanos al a destino). En la terminología estándar cuando se hablaba del algoritmo A*, $g(n)$ representa el costo acumulado de la trayectoria del origen a cualquier nodo n , y $h(n)$ representa el costo estimado por la heurística del nodo n a la punto destino. El algoritmo balancea los dos valores según se mueve desde el origen al destino y mantiene dos listas: una abierta y otra cerrada en donde la lista cerrada almacena los nodos que ya han sido evaluados, y la lista abierta contiene a los nodos que aún no los son. Cada vez que se inicia un nuevo ciclo del algoritmo se expande el nodo n que tiene la $f(n)$ más baja $f(n) = g(n) + h(n)$ (Tomer Livneh *et al.*, [15]).

Convenciones:

- $h(n)$ Heurística que estima el costo de n hasta el destino.
- $g(n)$ El costo conocido desde el origen hasta n . Este valor es actualizado durante el funcionamiento.
- $f(n) = h(n) + g(n)$ Estimación completa del costo de alcanzar desde el origen al destino a través del n . Este valor es actualizado durante el funcionamiento.

Entradas:

- Un grafo pesado $G = (V, E)$.
- n_i Un nodo origen.
- n_f Un nodo destino.

Salidas:

- Una trayectoria desde n_i hasta n_f (Si ésta existe).

El pseudocódigo del algoritmo A* implementado se muestra en el algoritmo 9.

Algoritmo 9 A-Estrella

Entrada: Espacio de configuraciones

Salida: Trayectoria de un n_i a n_f

$ListaCerrada \leftarrow \emptyset$

$ListaAbierta \leftarrow n_i$

$n \leftarrow n_x \in ListaAbierta \{n \text{ con valor mínimo de } f(n)\}$

$ListaCerrada \leftarrow ListaCerrada \cup n, ListaAbierta \leftarrow \setminus n$

$rotResol \leftarrow$ Resolución de rotación

$Cspace[vResol][hResol][rotResol] \leftarrow \emptyset$

$capas \leftarrow \frac{360}{rotResol}$

$columnas \leftarrow$ Tamaño horizontal ambiente / hResol

$filas \leftarrow$ Tamaño vertical ambiente / vResol

si $n = n_f$ **entonces**

regresa $solucion(n)$

fin si

$N_n \leftarrow$ Vecinos de n

para cada $n' \in N_n$ **hacer**

$padre(n') \leftarrow n$

$g'(n') \leftarrow g(n) + costo(n, n')$

$f'(n') \leftarrow g'(n') + h(n')$

si $n' = n_x \in ListaAbierta$ & $f'(n') \geq f(n_x)$ **entonces**

 Continuar

sino si $n' = n_x \in ListaCerrada$ **entonces**

 Continuar

sino

$ListaAbierta \leftarrow ListaAbierta \setminus copia \text{ de } n'$

$ListaAbierta \leftarrow ListaAbierta \cup n'$

fin si

fin para

regresa $solucion(n)$

4.6.1 Heurística A* modificada

Para efectos comparativos, se desarrolló una variante del algoritmo A* que, en lugar de optimizar la distancia (como es el caso del algoritmo original), busque además optimizar el número de cambios de dirección.

La modificación que se propone es el remplazo de la función $g(n)$, que ahora tendrá un costo asociado a la cantidad de cambios de dirección y no el del costo de distancia desde el origen hasta n . Con esta modificación pretendemos enfocar la búsqueda a la reducción en los cambios de dirección.

En la Figura 4.11 A se muestra una ruta obtenida con la heurística básica del algoritmo A* en la que se busca reducir la distancia, para este ejemplo esta trayectoria registro 18 cambios de dirección. En la Figura 4.11 B se muestre el mismo ejemplo, pero en éste se utiliza una heurística de reducción de cambios de dirección. Así, esta ruta obtuvo la menor cantidad de cambios de dirección (requiriendo sólo 9), pero obteniendo una ruta de mayor longitud.

4.7 Resumen del capítulo

En este capítulo se describieron las heurísticas propuestas para resolver el problema del cálculo de trayectorias reduciendo el cambio de direcciones de objeto, tomando en cuenta únicamente el aspecto geométrico del problema. Fenómenos físicos como la fricción y la inercia se considerarán en trabajos futuros.

También se presentó el algoritmo para la construcción del frente de onda que es la base del funcionamiento de las heurísticas propuestas. Finalmente, se presentó el método

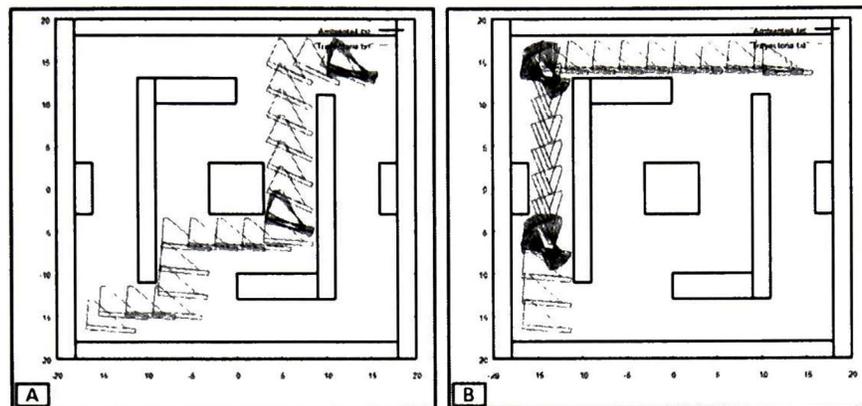


Figura 4.11: A) Ruta calculada con el algoritmo A* usando la heurística de reducción de distancia. B) Ruta calculada con el algoritmo A* usando la heurística de reducción de cambios de dirección.

utilizado para obtener los puntos de empuje. Los resultados preliminares que se muestran en el capítulo sugieren que las heurísticas desarrolladas en este trabajo de investigación son competentes al compararlas con estrategias clásicas para el cálculo de trayectorias.

5

Resultados

“Para 2050, se podrán desarrollar un equipo de robots humanoides de carácter autónomo que será capaz de ganar al equipo de fútbol campeón del mundo”

Fragmento tomado del sitio oficial de RoboCup¹

En este capítulo abordaremos los aspectos relacionados con la implementación final de los métodos propuestos. Igualmente, mostraremos y analizaremos los resultados obtenidos del proceso de pruebas que se realizó. Para ello, inicialmente se presentan los aspectos más relevantes sobre el diseño de las pruebas (descripción del entorno de trabajo, ambientes de pruebas, métricas, etc). Posteriormente se muestran los resultados finales de las pruebas y finalmente se presenta el análisis de los resultados obtenidos.

¹RoboCup es un proyecto internacional para promover, a través de competiciones integradas por robots autónomos, la investigación y educación sobre inteligencia artificial. <http://www.robocup.org/>

5.1 Esquemas a prueba

Las heurísticas propuestas en esta tesis han sido validadas experimentalmente mediante la realización de una serie de pruebas. En total, se implementaron las 3 heurísticas propuestas, así como 4 heurísticas clásicas para efectos comparativos.

A continuación se listan las 7 heurísticas implementadas:

1. Frente de onda con selección de vecino aleatoria (SVA)
2. Frente de onda con selección de vecino predefinida (SVP)
3. Frente de onda con medición de alcance (MA)
4. Frente de onda con medición de alcance y validación de capa (MAVC)
5. Frente de onda con construcción de grafo de ruta (GR)
6. A Estrella simple (A*S)
7. A Estrella con heurística de reducción de reconfiguraciones (A*RR)

Las heurísticas 1, 2 y 6 se orientan a la reducción de distancias y el resto a la reducción de reconfiguraciones.

Es importante mencionar que todos los algoritmos implementados son determinísticos² exceptuando al frente de onda con selección de vecino aleatoria "SVA".

En la siguiente sección se describe puntualmente el conjunto de experimentos diseñados para evaluar las capacidades de las heurísticas propuestas en esta tesis.

²Un algoritmo determinístico es un algoritmo que, en términos informales, es completamente predictivo si se conocen las entradas al mismo

5.2 Descripción de los experimentos.

El conjunto total de pruebas fue constituido por la construcción 2539 rutas aleatorias divididas en 5 ambientes de pruebas distintos y 2 resoluciones del espacio de configuraciones. La segmentación del conjunto de pruebas se presenta a continuación:

Clasificación por la resolución del C-space:

1. C-space con resolución de 50cm: 350 Rutas
2. C-space con resolución de 20cm: 2189 Rutas

Clasificación por heurística:

En la tabla 5.1 se presenta las cantidades de pruebas que se realizaron divididas por heurísticas. Debido a los tiempos de cómputo importantes, el conjunto de pruebas para las heurísticas "A*" y "A*RR" fue disminuido.

| Heurística \ Rutas | Cantidad |
|--------------------|----------|
| SVA | 450 |
| SVP | 450 |
| MA | 450 |
| MAVC | 450 |
| GR | 450 |
| A*S | 50 |
| A*RR | 239 |

Tabla 5.1: Cantidad de pruebas realizada por cada heurística

Clasificación por ambiente de pruebas (Tabla 5.2)

| Ambiente \ Rutas | Cantidad |
|--------------------------------|-----------------|
| 1 | 410 |
| 2 | 405 |
| 3 | 414 |
| 4 | 830 |
| 5 | 480 |

Tabla 5.2: Cantidad de pruebas realizada por cada ambiente

Clasificación por ambiente\heurística. Resolución 20cm (Tabla 5.3):

| Heurística \ Ambiente | 1 | 2 | 3 | 4 | 5 |
|-------------------------------------|----------|----------|----------|----------|----------|
| SVA | 80 | 80 | 80 | 80 | 80 |
| SVP | 80 | 80 | 80 | 80 | 80 |
| MA | 80 | 80 | 80 | 80 | 80 |
| MAVC | 80 | 80 | 80 | 80 | 80 |
| AR | 80 | 80 | 80 | 80 | 80 |
| A*S | 0 | 0 | 0 | 0 | 0 |
| A*RR | 10 | 5 | 14 | 80 | 80 |

Tabla 5.3: División de pruebas ambiente\heurística. Resolución 20cm

Por ambiente\heurística. Resolución 50cm (Tabla 5.4):

Debido a la cantidad de tiempo que se consumió en el cálculo de trayectorias basadas en el algoritmo de A Estrella, fue imposible realizar la misma cantidad de pruebas de estas heurísticas en comparación con el número de pruebas realizadas para las heurísticas basadas en el algoritmo del frente de onda.

| Heurística \ Ambiente | 4 |
|-----------------------|----|
| SVA | 50 |
| SVP | 50 |
| MA | 50 |
| MAVC | 50 |
| AR | 50 |
| AES | 50 |
| AERR | 50 |

Tabla 5.4: División de pruebas ambiente\heurística. Resolución 50cm

5.2.1 Ambientes

Para abarcar un amplio intervalo de posibilidades se han considerado espacios 5 diferentes espacios de trabajo que varían en su complejidad, cantidad de obstáculos y en el porcentaje de región libre. Estos espacios de trabajo tienen unas dimensiones de $40\text{ m} \times 40\text{ m}$ (para un área total de 1600m^2) y se trabajaron con dos distintas resoluciones, la primera de 20 cm por lado y pasos de rotación de 5° , lo que equivale a una matriz de $201 \times 201 \times 72$ que supone 2,908,872 voxeles³. La segunda resolución fue de 50 cm por lado y 5° para los pasos de rotación, para trabajar con una matriz $81 \times 81 \times 72$ que equivale a tener un espacio de búsqueda de 472,392 configuraciones.

Las características de los ambientes que se crearon para probar las heurísticas propuestas se listan a continuación (Tabla 5.5).

En las Figuras 5.1, 5.2, 5.3, 5.4 y 5.5 se muestran gráficamente los distintos ambientes de pruebas.

³Un voxel es la parte más pequeña distinguible con forma de cubo de una imagen tridimensional.

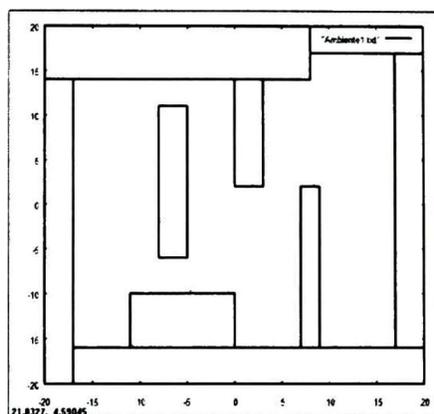


Figura 5.1: Ambiente de pruebas 1, con 9 obstáculos y un 44.4 % de área libre.

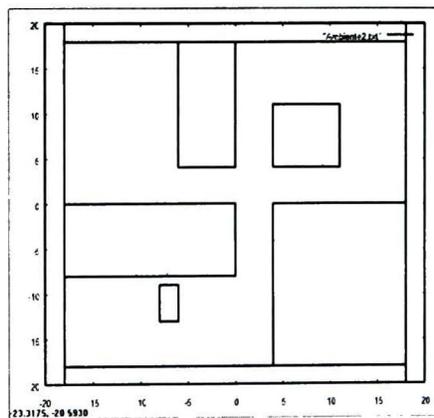


Figura 5.2: Ambiente de pruebas 2, con 9 obstáculos y un 49 % de área libre.

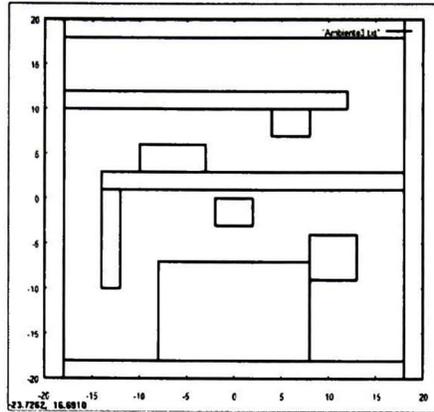


Figura 5.3: Ambiente de pruebas 3, con 12 obstáculos y un 56.4 % de área libre.

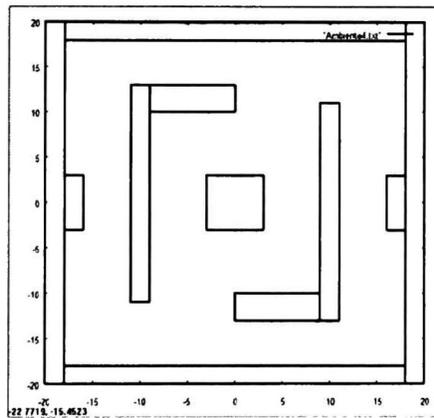


Figura 5.4: Ambiente de pruebas 4, con 11 obstáculos y un 67.9 % de área libre.

| Ambiente | 1 | 2 | 3 | 4 | 5 |
|-----------------------|----------|----------|----------|----------|----------|
| Característica | | | | | |
| Cantidad obstáculos | 9 | 9 | 12 | 11 | 12 |
| Área libre | 44,4 % | 49 % | 56,4 % | 67,9 % | 79 % |
| Área ocupada | 53,6 % | 51 % | 43,6 % | 32,1 % | 21 % |

Tabla 5.5: Características de los ambientes de pruebas

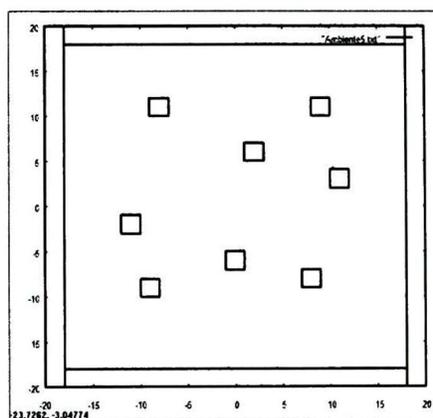


Figura 5.5: Ambiente de pruebas 5, con 12 obstáculos y un 79 % de área libre.

5.2.2 Métricas

Para lograr la caracterización adecuada del comportamiento de las heurísticas propuestas, se definieron los siguientes parámetros de medición:

1. Cantidad de reconfiguraciones: Posteriormente a la construcción de las rutas, se realiza un conteo de los cambios de dirección encontrados en la ruta. Este valor representa la cantidad de reconfiguraciones necesarias de los robots para efectuar la trayectoria.
2. Tiempo: Referente al tiempo máquina que requirió cada heurística para realizar la construcción de las rutas.

Las medidas estadísticas empleadas en el análisis de los resultados son las siguientes:

- Promedio
- Valor máximo
- Valor mínimo
- Desviación estándar
- Cantidad total

5.2.3 Ambiente de implementación

Las heurísticas fueron implementadas en el lenguaje de programación JAVA y las pruebas fueron realizadas en una máquina portátil HP Pavilion dv9000 de las siguientes características:

- Procesador: Intel Core 2 Duo T7200 a 2.00 GHz
- Memoria RAM: 2046 MB
- Sistema Operativo: Windows Vista 64-bit

5.3 Resultados

En esta sección se describen los resultados más significativos del total de pruebas realizadas. En la sección de Anexos se puede consultar los demás resultados.

5.3.1 Primera fase de pruebas

Por las dimensiones del espacio de búsqueda, no fue posible aplicar las heurísticas basadas en el algoritmo A* a los casos de resolución de 20cm. Por ello, los resultados

comparativos de las 7 heurísticas se obtuvieron para los casos con resolución de 50cm. A continuación se resumen los resultados obtenidos para el medio ambiente mostrado en la Figura 5.4. Cada heurística fue aplicada para resolver 50 problemas de planificación de trayectorias, con puntos iniciales y finales aleatorios.

5.3.1.1. Reconfiguraciones resolución 50cm

La tabla 5.6 muestra los resultados sobre el número de cambio de dirección de las 50 trayectorias obtenidas por cada heurística, indicando valores máximo, mínimo y promedio para cada una de ellas. Es importante recordar que, para nuestro enfoque sobre el problema de planificación, es más atractiva una heurística que genere trayectorias con un número reducido de cambios de dirección.

| Heurística \ Estadística | Promedio | Máximo | Mínimo | Desviación Est. |
|--------------------------|----------|--------|--------|-----------------|
| SVA | 22.08 | 51 | 4 | 11.02 |
| A* | 13.1 | 40 | 3 | 8.65 |
| SVP | 12.16 | 27 | 2 | 8.06 |
| A*RR | 6.98 | 14 | 3 | 2.29 |
| MAVC | 5.94 | 13 | 2 | 2.4 |
| MA | 5.84 | 12 | 2 | 2.39 |
| GR | 5.2 | 10 | 2 | 1.92 |

Tabla 5.6: Reconfiguraciones por heurística. Ambiente 4. Resolución 50cm

En estos resultados, resulta claro que las heurísticas (SVA, SVP y A*) fueron superadas ampliamente por el resto de las heurísticas. Sin embargo, esta situación es congruente con el hecho de que dichas heurísticas no incluyen ninguna estrategia que busque reducir el número de cambios de dirección, lo que es el caso de las otras 4 metodologías. Es importante destacar que la heurística "GR" obtuvo los mejores

resultados totales, sin embargo, fue superada por la heurística "A*RR" para una situación en específico (Ver Apéndice). El análisis de un caso similar a éste es presentado en la sección 5.4.

En la Figura 5.6 se muestra la gráfica correspondiente a la tabla 5.6.

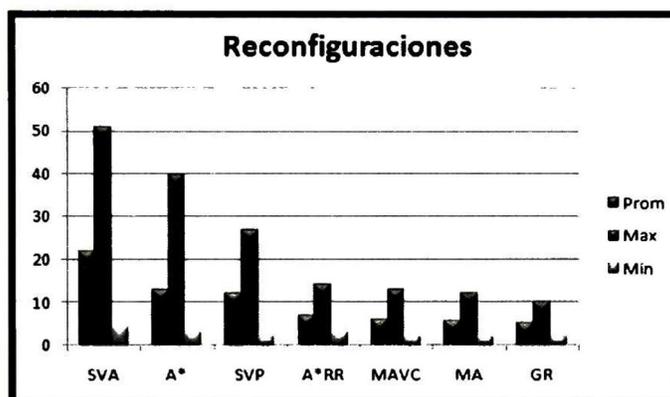


Figura 5.6: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 50cm

5.3.1.2. Tiempo resolución 50cm

En la tabla 5.7 se presentan los datos referentes a las tiempos de cálculo. Los valores de tiempo en la tabla están representados en milisegundos e indican el tiempo máquina necesario para construir la ruta.

Como se puede ver en la tabla de tiempos (Tabla 5.7), los mejores tiempos de cálculo corresponden a las heurísticas "SVP", "SVA", "MA" y "MAVC". Sin embargo, los tiempos de la heurística "GR" fueron muy competentes y son claramente compensados cuando se considera el número de cambios de dirección mostrados en la tabla 5.6.

| Heurística \ Estadística | Promedio | Máximo | Mínimo | Desviación Est. |
|--------------------------|----------|--------|--------|-----------------|
| SVP | 375 | 375 | 375 | 0 |
| SVA | 375.08 | 377 | 375 | 0.34 |
| MA | 375.02 | 376 | 375 | 0.14 |
| MAVC | 375.02 | 376 | 375 | 0.14 |
| GR | 421 | 438 | 406 | 9.78 |
| A* | 24948.3 | 272625 | 1 | 53531.45 |
| A*RR | 1174.34 | 10062 | 1 | 2109.2 |

Tabla 5.7: Tiempo por heurística. Ambiente 4. Resolución 50cm. Tiempos expresados en *ms*

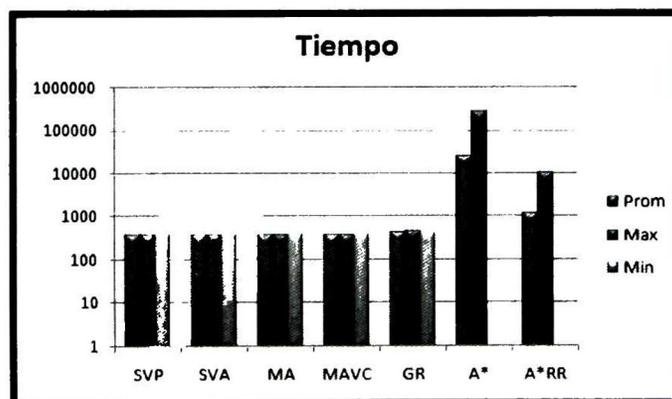


Figura 5.7: Gráfica de tiempo por heurística. Ambiente 4. Resolución 50cm. Tiempos expresados en *ms*

5.3.2 Segunda fase de pruebas

El objetivo de esta fase es estudiar el desempeño de las heurísticas propuestas, cuando son confrontadas con un espacio de búsqueda más grande, al aumentar la resolución del espacio de configuraciones.

Al estudiar los casos de pruebas con resoluciones más altas, los tiempos de cálculo requeridos por la heurística "A*S" se volvieron prohibitivos y, al no ser compatibles con una noción de respuesta en tiempo real, se decidió eliminarla del estudio comparativo.

Para la segunda fase de pruebas se utilizó una resolución más alta para la construcción del espacio de configuraciones (20 cm) y el conjunto de pruebas se extendió a la construcción de 80 rutas entre posiciones iniciales y finales tomadas aleatoriamente. Por motivos de tiempo de cómputo, se excluyó de las pruebas al algoritmo A* simple. Los ambientes utilizados en esta segunda fase fueron los ambientes 4 y 5. Las tablas y gráficas de este conjunto de prueba pueden ser consultados a continuación (Tabla 5.8, Tabla 5.9 y gráfica 5.8).

Las conclusiones de esta fase son similares a las de la primera fase, con los algoritmos "GR" y "A*RR" que proporcionan las trayectorias con un menor número de cambios de dirección, pero el algoritmo "A*RR", aunque puede encontrar trayectorias mejores que el "GR", requiere de tiempos de cómputo más importantes.

5.3.2.1. Reconfiguraciones resolución 20cm

| Heurística \ Estadística | Promedio | Máximo | Mínimo | Desviación Est. |
|--------------------------|----------|--------|--------|-----------------|
| SVA | 40.87 | 108 | 2 | 22.66 |
| SVP | 19.87 | 62 | 1 | 15.75 |
| A*RR | 8.31 | 13 | 2 | 2.75 |
| MAVC | 6.36 | 18 | 1 | 3.42 |
| MA | 6.28 | 19 | 1 | 3.5 |
| GR | 5.66 | 16 | 1 | 2.92 |

Tabla 5.8: Reconfiguraciones por heurística. Ambiente 4. Resolución 20cm

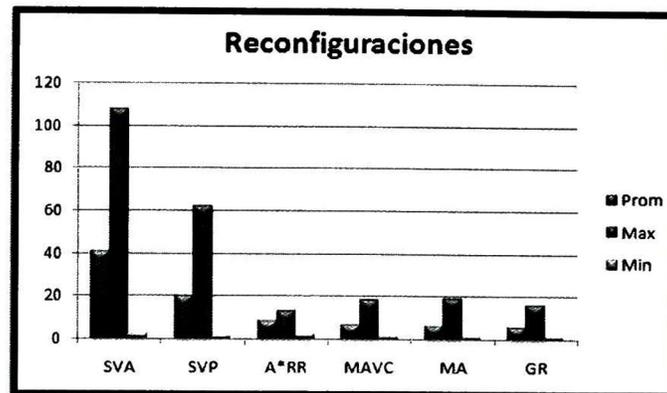


Figura 5.8: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 20cm

5.3.2.2. Tiempo resolución 50cm

| Heurística \ Estadística | Promedio | Máximo | Mínimo | Desviación Est. |
|--------------------------|----------|--------|--------|-----------------|
| SVP | 2996.23 | 2997 | 2996 | 0.42 |
| SVA | 2996.4 | 2997 | 2996 | 0.51 |
| MA | 2996.2 | 2997 | 2996 | 0.4 |
| MAVC | 2996.37 | 3004 | 2996 | 1.02 |
| GR | 3056.8 | 3111 | 3042 | 11.33 |
| A*RR | 22082.1 | 148510 | 1 | 42387.6 |

Tabla 5.9: Tiempo por heurística. Ambiente 4. Resolución 20cm. Tiempos expresados en ms

5.3.3 Tercera fase de pruebas

En esta tercera fase, se confrontó a las heurísticas con medios ambientes más complejos, con corredores estrechos que restringen de manera más significativa los posibles desplazamientos del objeto.

Debido al incremento en la complejidad de los ambientes considerados, los tiempos de computo del algoritmo "A*RR" aumentaron de manera muy notable, lo que lo volvió incompatible con un esquema de planeación que se desea capaz de responder rápidamente, por lo que no fue considerado en esta fase de pruebas. Los ambiente utilizado en esta fase son el 1 , 2 y 3.

Es necesario resaltar, que aunque sólo se presentan los resultados de las heurísticas basadas en el frente de onda, el experimento incluyo algunas pruebas con el algoritmo A* con la heurística de reducción de reconfiguraciones. Estos resultados pueden ser consultados en la sección de apéndices.

5.4 Casos específicos.

En esta sección se discuten las situaciones más sobresalientes que fueron observadas durante el proceso de experimentación.

El primer caso de prueba que merece ser detalladamente analizado es cuando el algoritmo "A*RR" es capaz de encontrar mejores soluciones, en el número de cambios de dirección que las otras heurísticas propuestas.

La ruta del ejemplo que será analizado se muestra en la Figura 5.9. La trayectoria de la izquierda fue calculada utilizando la heurística de "Grafo de ruta", y requiere 10 cambios de dirección. Por otro lado tenemos la ruta de la derecha que fue calculada con el algoritmo $A * RR$, requiriendo solamente 9 cambios de dirección.

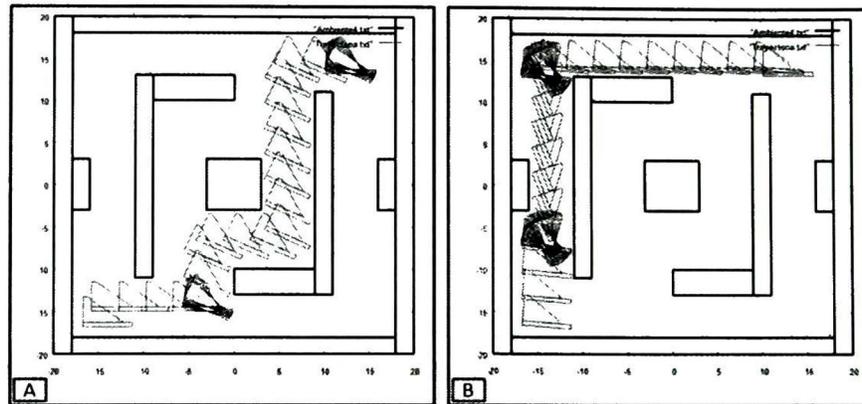


Figura 5.9: La ruta A

Revisando detalladamente este ejemplo, se encontró que el motivo por el cual la heurística "Grafo de ruta" fue superada es la siguiente: Como se mencionó en el capítulo 4, el algoritmo de frente de onda es un mecanismo utilizado para reconstruir rutas que unan 2 puntos, pero con las características adicionales de que este método puede evitar obstáculos, mínimos locales y encontrar rutas que se acercan continuamente a la posición final. Esta última característica es la que causante de que la heurística "Grafo de ruta" fuera superada.

El algoritmo de "frente de onda" fue inspirado por el comportamiento de la propagación de ondas esféricas, y al aplicar este mecanismo en un ambiente delimitado inevitablemente se llegará a un punto en que los distintos frentes de propagación se encontrarán unos con otros.

Revisando a detalle la matriz marcada con el algoritmo del frente de onda se pudo observar que cada uno de estos choques crea una "división" en la que los valores de cada lado de la división impiden que el seguimiento descendente de los nodos continúe en el proceso de construcción de rutas y por lo tanto se impida el avance en la dirección del

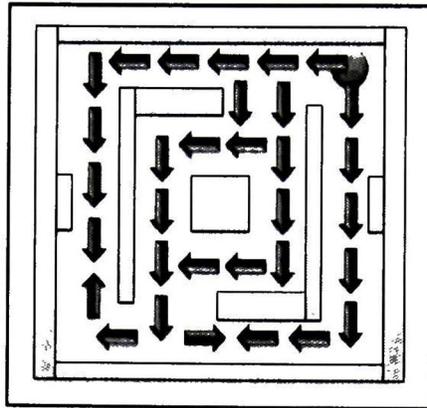


Figura 5.10: Ejemplo de dos frentes de onda que se encuentran entre ellos.

choque de los frentes. En la Figura 5.10 se muestra un ejemplo en el que los diferentes frentes de propagación se encuentran y generan una de estas "divisiones".

El algoritmo A^* con la heurística de reducción de reconfiguraciones fue capaz de obtener un mejor resultado en cuanto a cambios de dirección, debido a que este método no busca necesariamente acercarse continuamente del punto final, sino solamente reducir el número de cambios de dirección. Esta "ventaja" le permite superar a la heurística "Grafo de ruta" en aquellas situaciones donde la *ruta con menos cambios de dirección no forma parte de las rutas más cortas en longitud*. Por esta razón, el algoritmo " A^*RR " puede realizar movimientos que alejen al robot del punto objetivo, y buscar trayectorias más largas pero con un menor cambio de direcciones. Sin embargo, el costo en esfuerzo computacional es mucho más importante.

Otra situación muy interesante es el caso en que los tiempos de cálculo de los algoritmos " A^* " y " A^*RR " son menores que las demás heurísticas (Tabla 5.7).

Esta situación se presenta cuando las posiciones inicial y final son muy próximas entre

sí, por lo que la ruta que se busca tiene una longitud muy corta. En estas condiciones, el cálculo del frente de onda sobre todo el espacio de configuraciones requiere de un tiempo de cómputo más importante que las búsquedas locales realizadas por las heurísticas "A*" y "A*RR".

5.5 Resumen de resultados

Acorde a los resultados que fueron presentados, podemos concluir que, aún cuando existen situaciones en las que el algoritmo A*RR puede obtener mejores resultados (trayectorias con menor número de cambios de dirección), las tres heurísticas propuestas en esta tesis ("MA", "MAVC" y muy particularmente la heurística "GR") obtienen, en general, un mejor compromiso entre calidad de la solución encontrada y esfuerzo computacional requerido.

Como era de esperarse, el refinamiento de cada heurística propuesta, partiendo desde la heurística "MA", permitió crear una heurística "GR" muy competente comparada con la heurística basada en el algoritmo A*.

Es importante destacar que la especialización de la heurística "GR" conlleva un ligero incremento en los tiempos de cómputo. Sin embargo, podemos concluir que, en general, la relación entre cantidad de reconfiguraciones, longitud de trayectorias y tiempo de cálculo beneficia claramente a la heurística "Grafo de ruta".

5.6 Resumen del capítulo

En este capítulo se abordaron los temas relacionados a la experimentación de la metodología propuesta en esta tesis, se describió el modelo de pruebas que fue

implementado y por supuesto se presentó el análisis correspondiente de los datos obtenidos, el análisis de las situaciones relevantes y las conclusiones de los análisis.

6

Conclusiones y trabajo futuro.

“Actualmente, las capacidades intelectuales que tienen los robots pueden compararse con las de un insecto. A lo mejor, en diez años se consigan máquinas con el intelecto de un ratón; los más optimistas consideran que en unos cincuenta años se podrán conseguir robots semejantes a un ser humano”

Michio Kaku¹

En este documento se ha descrito el desarrollo de mi trabajo de investigación, desde las motivaciones del proyecto hasta las pruebas realizadas y los resultados obtenidos de las mismas.

El objetivo de este trabajo era desarrollar una estrategia original para resolver el problema del movimiento de objetos por medio de comunidades de robots móviles, incluyendo la evasión de colisiones con los obstáculos del medio ambiente no estructurado.

¹Declaración hecha al periódico 'La Nación de Argentina' el profesor de física Michio Kaku, de la City University de Nueva York en el 2003.

El conjunto de hipótesis considerado resulta adecuado para obtener los primeros resultados del enfoque propuesto y cubrir los objetivos planteados inicialmente. Sin duda será necesario, en trabajos futuros, abrir el conjunto de hipótesis para incluir algunos aspectos físicos no considerados, con el objeto de mejorar la propuesta.

El principal aporte de este trabajo de investigación consiste en el desarrollo de una estrategia novedosa de planificación de movimientos, diseñada para construir rutas válidas para desplazar un objeto de un punto a otro en ambientes no estructurados, por medio de robots móviles no especializados. Como resultado, se proponen 3 estrategias heurísticas que, además de realizar la planificación de movimientos del objeto a desplazar, obtienen trayectorias óptimas o cercanas al óptimo que reducen el tiempo de traslado del objeto, al reducir el número de cambios de dirección durante el desplazamiento. Cabe recordar que, al tratarse de una tarea donde a los robots móviles sólo les está permitido empujar el objeto, cambiar la dirección de movimiento requiere reubicar los robots alrededor del objeto, lo que requiere de un tiempo de ejecución.

A pesar de que el movimiento de objetos por medio de comunidades de robots móviles es un problema ampliamente estudiado por la comunidad científica en robótica, existen muy pocos trabajos que consideren el desplazamiento en medios ambientes no estructurados. Esta observación se asimiló como un área de oportunidad y definió el curso del proyecto.

La propuesta desarrollada parte de una descripción métrica del medio ambiente, para obtener una representación discreta del espacio de configuraciones del objeto a desplazar. Por medio de un algoritmo de frente de onda, se obtiene una descripción del espacio solución al problema de planificación de trayectorias para el objeto. Este espacio

solución tiene la característica de que posee un único mínimo global, por lo que permite desarrollar estrategias de optimización en la búsqueda de soluciones, sin que el algoritmo de optimización pueda quedar atrapado en un mínimo local. Al construir la trayectoria en esta representación, se obtienen dos ventajas importantes: por un lado, al construir la solución en el espacio de configuraciones de la caja, se garantiza que no existen colisiones entre el objeto y los obstáculos del medio ambiente; por otra parte, se pueden obtener trayectorias que optimicen algún criterio dado.

En nuestro caso, el criterio a minimizar fue el número de cambios de dirección de la trayectoria obtenida. Se trata de un problema combinatorio bastante complejo, cuyo espacio solución crece rápidamente con la longitud total de la trayectoria, ya que cada paso de la trayectoria tiene más de una opción para el siguiente desplazamiento. Por esta razón, se propone el uso de una heurística para resolverlo, que evolucionó naturalmente desde un algoritmo voráz (denominado "Medición de Alcance") hasta un algoritmo de construcción de mejores rutas ("Grafo de Rutas").

Para obtener una correcta dimensión de la eficiencia de nuestra propuesta, se realizó un estudio comparativo con una de las técnicas clásicas para la planificación de trayectorias más utilizadas en robótica móvil: el algoritmo A*, y una modificación a este algoritmo diseñada para optimizar el mismo criterio que el de nuestra propuesta.

A través de los resultados de la experimentación realizada, es posible comprobar que el método propuesto logra cumplir con los objetivos planteados al inicio del proyecto, y ofrece un excelente compromiso entre la calidad de la solución obtenida y el esfuerzo computacional requerido para calcularla.

La fase de experimentación permitió también definir las limitaciones prácticas de la estrategia propuesta:

- El método propuesto sólo puede construir rutas en las que los movimientos siempre son para acercar el objeto al destino. Esta característica favorece la construcción de rutas cortas, pero impide encontrar rutas que, a pesar de ser de mayor longitud, requieran un menor número de cambios de dirección de desplazamiento.
- El utilizar el centro de masa del objeto como centro de rotación fijo, limita las posibilidades de realizar ciertos movimientos del objeto que en determinados casos podrían reducir las distancias de las rutas, esto es, no se consideran otros centros de rotación que facilitarían los movimientos del objeto en algunas condiciones.
- La calidad de las soluciones encontradas está relacionada directamente con la resolución del espacio de configuraciones, sin embargo, utilizar resoluciones muy elevadas incrementan además de los tiempos las necesidades de almacenamiento y procesamiento.

6.1 Trabajo futuro

A partir de los resultados de este trabajo de tesis, se pueden plantear algunos puntos en los que es posible extender el trabajo presentado y que dan una idea de la importancia de esta línea de investigación para el futuro. A continuación se listan estas líneas de trabajo y se dan algunas ideas acerca de cómo pueden ser abordadas.

- Los puntos, sobre el perímetro del objeto, que son válidos para empujar en un momento dado cambian a lo largo de la trayectoria, dependiendo de la dirección en que se requiere desplazar el objeto. A cada cambio de dirección, donde los robots necesitan reacomodarse alrededor del objeto, es posible seleccionar puntos

de empuje que sean válidos no sólo para el segmento actual, sino también para el siguiente. Esto se reflejaría como un robot que no necesita desplazarse durante la reconfiguración. La selección de los puntos de empuje más ventajosos es entonces un problema de optimización combinatoria muy interesante, que permitiría reducir los tiempos requeridos para las reconfiguraciones.

- Relacionado con el punto anterior, se propone que la selección de los puntos de empuje sea complementada con una función de costo que evalúe la conveniencia de mover los robots de una posición a otra. Es decir, podría presentarse una situación donde dos de tres robots no requieren reacomodarse, pero el tercero debe rodear el objeto para colocarse en su posición. En este caso podría ser más conveniente desplazar todos los robots simultáneamente una distancia corta que un sólo robot una distancia larga. Esta técnica podría contribuir aún más en la reducción de tiempos de reconfiguración de los robots y por lo tanto impactar en los tiempos globales del sistema.
- Como se describió en el capítulo 5, utilizar una heurística basada en una matriz de frente de onda garantiza obtener las rutas más cortas, sin embargo, limita la búsqueda de soluciones que pueden ser mejores (a costa de longitudes mayores). Para resolver este problema, proponemos el desarrollo de un algoritmo que comparta la característica del frente de onda de poseer un único mínimo global, pero enfocado a contabilizar y reducir los cambios de dirección en lugar de las distancias.
- Durante el desplazamiento del objeto, no se requiere que todos los robots participen de la tarea de empuje, sobre todo si se cuenta con un número importante de robots. Una distribución ventajosa en el medio ambiente de los robots que no formen parte activa de un movimiento de empuje, puede contribuir aún más en la reducción de tiempos de traslado. Esto se debe a que los robots pueden trasladarse a posiciones

que faciliten y agilizen los movimientos posteriores de la caja.

- Para este trabajo de tesis sólo se consideraron ambientes planos, con desplazamientos horizontales. Como trabajo futuro se propone llevar el modelo propuesto a un problema de desplazamiento tridimensional del objeto, utilizando robots especializados en manipulación de objetos.
- Finalmente, sería interesante llevar a cabo la implementación de esta estrategia en robots reales, por lo que habría que incluir muchos aspectos físicos que, para el desarrollo de esta tesis, se han considerado ideales.

A

Resultados

A continuación se muestran los resultados obtenidos de la fases de pruebas 3, 2 y 1 respectivamente. Los datos mostrados de los tiempos de cálculo para las heurísticas "SVP", "SVA", "MA", "MAVC" y "GR", son sólo referentes al tiempo de construcción de la ruta, por lo tanto a ese cantidad se le tiene que adicionar el tiempo de la propagación del frente de onda.

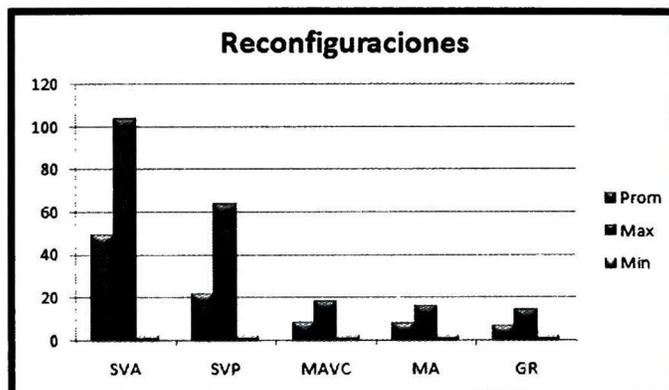


Figura A.1: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 1. Resolución 20cm. Ver Tablas A.1 y A.2

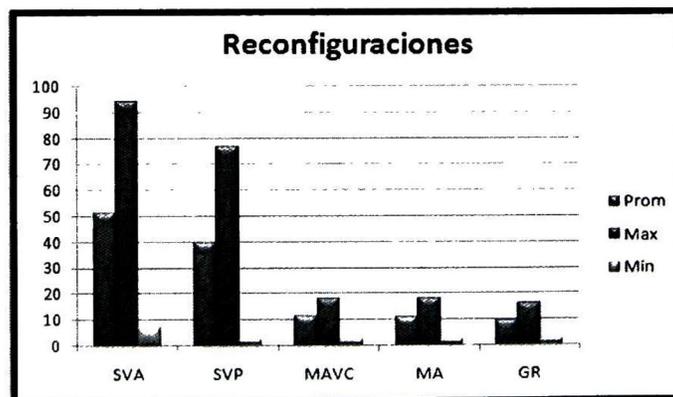


Figura A.2: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 2. Resolución 20cm. Ver Tablas A.3 y A.4

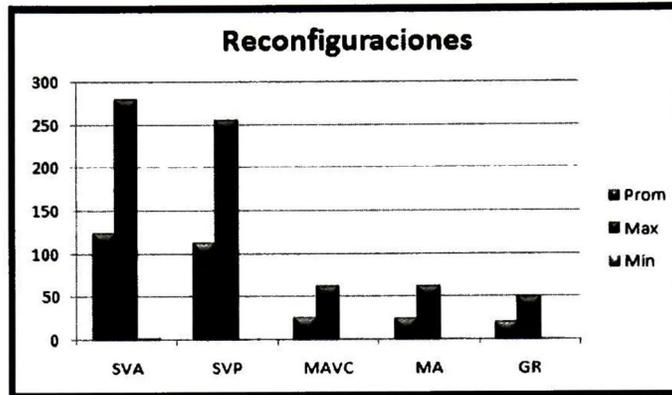


Figura A.3: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 3. Resolución 20cm. Ver Tablas A.5 y A.6

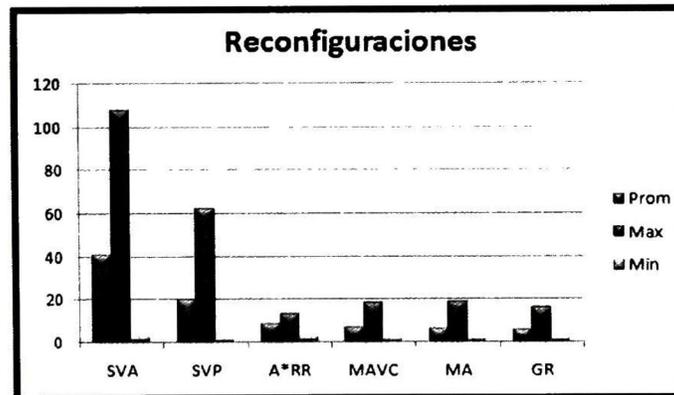


Figura A.4: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 20cm. Ver Tablas A.7 y A.8

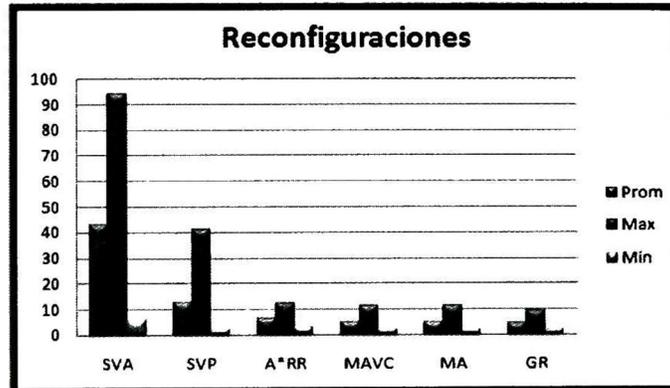


Figura A.5: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 5. Resolución 20cm. Ver Tablas A.9 y A.10

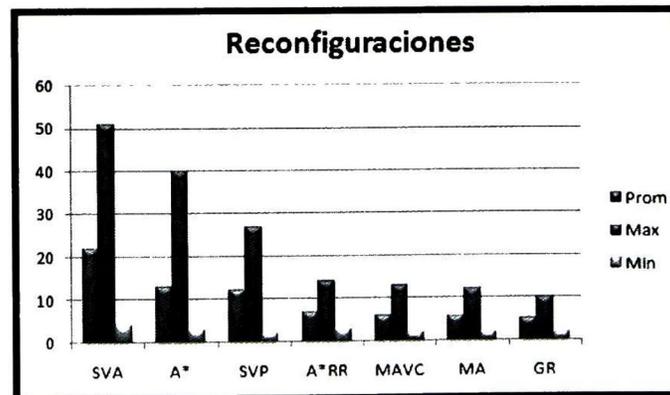


Figura A.6: Gráfica de cantidad de reconfiguraciones por heurística. Ambiente 4. Resolución 50cm. Ver Tablas A.11, A.12 y A.13

Tabla A.1: Ambiente 1. Resolución: 20cm. Propagación de frente de onda: 4305ms

| Prueba | SVP | | | SVA | | | MA | | |
|----------------|-------------------|-----------------|-----------------|-------------------|-----------------|-----------------|-------------------|-----------------|-----------------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 3 | 40 | 0 | 16 | 40 | 0 | 3 | 40 | 1 |
| 2 | 44 | 137 | 0 | 55 | 137 | 0 | 10 | 137 | 0 |
| 3 | 19 | 124 | 0 | 56 | 124 | 1 | 7 | 124 | 0 |
| 4 | 44 | 121 | 0 | 62 | 121 | 0 | 9 | 121 | 0 |
| 5 | 15 | 133 | 0 | 46 | 133 | 0 | 10 | 133 | 0 |
| 6 | 31 | 196 | 0 | 51 | 196 | 0 | 9 | 196 | 0 |
| 7 | 2 | 27 | 0 | 2 | 27 | 0 | 2 | 27 | 0 |
| 8 | 19 | 163 | 0 | 30 | 163 | 0 | 6 | 163 | 0 |
| 9 | 16 | 176 | 0 | 29 | 176 | 0 | 6 | 176 | 0 |
| 10 | 26 | 119 | 0 | 60 | 119 | 0 | 11 | 119 | 0 |
| 11 | 2 | 12 | 0 | 4 | 12 | 0 | 2 | 12 | 0 |
| 12 | 42 | 212 | 0 | 62 | 212 | 2 | 7 | 212 | 0 |
| 13 | 32 | 160 | 0 | 68 | 160 | 1 | 10 | 160 | 0 |
| 14 | 48 | 123 | 0 | 73 | 123 | 0 | 15 | 123 | 0 |
| 15 | 28 | 142 | 0 | 58 | 142 | 0 | 10 | 142 | 0 |
| 16 | 21 | 118 | 0 | 47 | 118 | 2 | 11 | 118 | 0 |
| 17 | 21 | 147 | 0 | 80 | 147 | 1 | 9 | 147 | 0 |
| 18 | 3 | 33 | 0 | 15 | 33 | 0 | 3 | 33 | 0 |
| 19 | 33 | 166 | 0 | 65 | 166 | 0 | 8 | 166 | 0 |
| 20 | 52 | 161 | 0 | 88 | 161 | 0 | 15 | 161 | 0 |
| 21 | 28 | 154 | 0 | 47 | 154 | 0 | 8 | 154 | 0 |
| 22 | 35 | 117 | 1 | 45 | 117 | 0 | 8 | 117 | 0 |
| 23 | 4 | 48 | 0 | 25 | 48 | 0 | 4 | 48 | 0 |
| 24 | 3 | 56 | 0 | 30 | 56 | 0 | 3 | 56 | 0 |
| 25 | 11 | 166 | 0 | 68 | 166 | 0 | 9 | 166 | 0 |
| 26 | 40 | 135 | 0 | 54 | 135 | 2 | 12 | 135 | 0 |
| 27 | 31 | 152 | 0 | 76 | 152 | 0 | 9 | 152 | 0 |
| 28 | 32 | 194 | 1 | 82 | 194 | 0 | 11 | 194 | 0 |
| 29 | 23 | 135 | 0 | 65 | 135 | 0 | 11 | 135 | 0 |
| 30 | 28 | 144 | 0 | 66 | 144 | 0 | 11 | 144 | 0 |
| 31 | 16 | 125 | 0 | 53 | 125 | 0 | 11 | 125 | 0 |
| 32 | 31 | 179 | 0 | 62 | 179 | 0 | 6 | 179 | 0 |
| 33 | 11 | 116 | 0 | 63 | 116 | 0 | 9 | 116 | 0 |
| 34 | 15 | 196 | 0 | 68 | 196 | 0 | 10 | 196 | 0 |
| 35 | 3 | 73 | 0 | 34 | 73 | 0 | 3 | 73 | 0 |
| 36 | 4 | 28 | 0 | 17 | 28 | 0 | 4 | 28 | 0 |
| 37 | 29 | 200 | 0 | 39 | 200 | 2 | 7 | 200 | 0 |
| 38 | 11 | 176 | 0 | 57 | 176 | 0 | 9 | 176 | 0 |
| 39 | 3 | 11 | 0 | 5 | 11 | 0 | 3 | 11 | 0 |
| 40 | 3 | 33 | 0 | 23 | 33 | 0 | 3 | 33 | 0 |
| 41 | 28 | 134 | 0 | 84 | 134 | 0 | 10 | 134 | 0 |
| 42 | 64 | 157 | 0 | 100 | 157 | 2 | 15 | 157 | 0 |
| 43 | 9 | 99 | 0 | 41 | 99 | 0 | 9 | 99 | 0 |
| 44 | 3 | 44 | 0 | 16 | 44 | 0 | 3 | 44 | 0 |
| 45 | 26 | 151 | 0 | 72 | 151 | 0 | 11 | 151 | 0 |
| 46 | 20 | 172 | 0 | 70 | 172 | 0 | 12 | 172 | 0 |
| 47 | 44 | 140 | 0 | 64 | 140 | 0 | 12 | 140 | 0 |
| 48 | 31 | 154 | 0 | 51 | 154 | 0 | 5 | 154 | 0 |
| 49 | 3 | 29 | 0 | 24 | 29 | 0 | 3 | 29 | 0 |
| 50 | 21 | 139 | 0 | 45 | 139 | 0 | 6 | 139 | 0 |
| 51 | 33 | 176 | 0 | 57 | 176 | 0 | 12 | 176 | 0 |
| 52 | 17 | 147 | 0 | 39 | 147 | 0 | 6 | 147 | 0 |
| 53 | 4 | 50 | 0 | 27 | 50 | 0 | 4 | 50 | 0 |
| 54 | 22 | 143 | 0 | 70 | 143 | 0 | 12 | 143 | 0 |
| 55 | 4 | 54 | 0 | 37 | 54 | 0 | 4 | 54 | 0 |
| 56 | 22 | 133 | 0 | 61 | 133 | 0 | 13 | 133 | 0 |
| 57 | 27 | 159 | 0 | 104 | 159 | 2 | 12 | 159 | 0 |
| 58 | 56 | 189 | 0 | 73 | 189 | 0 | 16 | 189 | 0 |
| 59 | 49 | 134 | 0 | 59 | 134 | 0 | 10 | 134 | 0 |
| 60 | 32 | 157 | 0 | 52 | 157 | 0 | 10 | 157 | 0 |
| 61 | 2 | 28 | 0 | 14 | 28 | 0 | 2 | 28 | 0 |
| 62 | 15 | 126 | 0 | 53 | 126 | 0 | 10 | 126 | 0 |
| 63 | 15 | 106 | 0 | 42 | 106 | 1 | 10 | 106 | 0 |
| 64 | 27 | 155 | 0 | 47 | 155 | 0 | 8 | 155 | 0 |
| 65 | 45 | 129 | 0 | 58 | 129 | 0 | 9 | 129 | 0 |
| 66 | 4 | 39 | 0 | 25 | 39 | 0 | 4 | 39 | 0 |
| 67 | 22 | 134 | 0 | 79 | 134 | 0 | 11 | 134 | 0 |
| 68 | 40 | 195 | 0 | 55 | 195 | 0 | 10 | 195 | 0 |
| 69 | 13 | 177 | 0 | 69 | 177 | 0 | 8 | 177 | 0 |
| 70 | 2 | 27 | 0 | 7 | 27 | 0 | 2 | 27 | 0 |
| 71 | 3 | 43 | 0 | 25 | 43 | 0 | 3 | 43 | 0 |
| 72 | 4 | 37 | 0 | 22 | 37 | 0 | 4 | 37 | 0 |
| 73 | 34 | 127 | 0 | 46 | 127 | 0 | 8 | 127 | 0 |
| 74 | 21 | 159 | 0 | 87 | 159 | 0 | 7 | 159 | 0 |
| 75 | 3 | 48 | 0 | 34 | 48 | 0 | 4 | 48 | 0 |
| 76 | 54 | 142 | 0 | 73 | 142 | 0 | 15 | 142 | 0 |
| 77 | 3 | 46 | 0 | 13 | 46 | 0 | 3 | 46 | 0 |
| 78 | 13 | 163 | 0 | 36 | 163 | 0 | 6 | 163 | 0 |
| 79 | 32 | 155 | 0 | 67 | 155 | 0 | 11 | 155 | 0 |
| 80 | 4 | 38 | 0 | 18 | 38 | 0 | 4 | 38 | 0 |
| Total | 1733 | 9583 | 2 | 3962 | 9583 | 16 | 638 | 9583 | 1 |
| Prom | 21.6625 | 119.7875 | 4305.025 | 49.525 | 119.7875 | 4305.2 | 7.975 | 119.7875 | 4305.013 |
| ValMin | 2 | 11 | 0 | 2 | 11 | 0 | 2 | 11 | 1 |
| ValMax | 64 | 212 | 1 | 104 | 212 | 2 | 16 | 212 | 0 |
| DesvEst | 15.78169 | 55.41208 | 0.15711 | 23.40452 | 55.41208 | 0.560289 | 3.687045 | 55.41208 | 0.111803 |

Tabla A.2: Ambiente 1. Resolución: 20cm. Propagación de frente de onda: 4305ms

| Prueba | MAVC | | | GR | | | A*RR | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 3 | 40 | 0 | 3 | 40 | 63 | 6 | 64 | 93 |
| 2 | 11 | 137 | 0 | 9 | 137 | 63 | 12 | 139 | 552056 |
| 3 | 8 | 124 | 0 | 6 | 124 | 47 | 14 | 150 | 152568 |
| 4 | 10 | 121 | 0 | 6 | 121 | 62 | 10 | 129 | 107360 |
| 5 | 10 | 133 | 0 | 6 | 133 | 62 | 11 | 135 | 222971 |
| 6 | 8 | 196 | 0 | 8 | 196 | 62 | 10 | 228 | 3564750 |
| 7 | 2 | 27 | 0 | 2 | 27 | 78 | 3 | 27 | 68 |
| 8 | 6 | 163 | 0 | 6 | 163 | 47 | 11 | 231 | 2255202 |
| 9 | 6 | 176 | 0 | 6 | 176 | 62 | 12 | 236 | 5356091 |
| 10 | 12 | 119 | 0 | 7 | 119 | 63 | 10 | 127 | 2839 |
| 11 | 2 | 12 | 0 | 2 | 12 | 62 | | | |
| 12 | 7 | 212 | 0 | 6 | 212 | 47 | | | |
| 13 | 10 | 160 | 0 | 7 | 160 | 47 | | | |
| 14 | 15 | 123 | 0 | 13 | 123 | 62 | | | |
| 15 | 10 | 142 | 0 | 7 | 142 | 47 | | | |
| 16 | 11 | 118 | 0 | 10 | 118 | 47 | | | |
| 17 | 11 | 147 | 0 | 9 | 147 | 47 | | | |
| 18 | 3 | 33 | 0 | 3 | 33 | 62 | | | |
| 19 | 8 | 166 | 0 | 7 | 166 | 63 | | | |
| 20 | 11 | 161 | 0 | 7 | 161 | 78 | | | |
| 21 | 8 | 154 | 2 | 8 | 154 | 78 | | | |
| 22 | 10 | 117 | 0 | 8 | 117 | 125 | | | |
| 23 | 4 | 48 | 0 | 4 | 48 | 63 | | | |
| 24 | 3 | 56 | 2 | 3 | 56 | 78 | | | |
| 25 | 9 | 166 | 0 | 7 | 166 | 62 | | | |
| 26 | 12 | 135 | 0 | 9 | 135 | 47 | | | |
| 27 | 8 | 152 | 0 | 8 | 152 | 47 | | | |
| 28 | 10 | 194 | 0 | 8 | 194 | 47 | | | |
| 29 | 12 | 135 | 0 | 11 | 135 | 62 | | | |
| 30 | 12 | 144 | 0 | 11 | 144 | 63 | | | |
| 31 | 11 | 125 | 0 | 10 | 125 | 47 | | | |
| 32 | 8 | 179 | 0 | 5 | 179 | 62 | | | |
| 33 | 9 | 116 | 0 | 7 | 116 | 62 | | | |
| 34 | 10 | 196 | 0 | 9 | 196 | 63 | | | |
| 35 | 3 | 73 | 0 | 3 | 73 | 62 | | | |
| 36 | 4 | 28 | 0 | 4 | 28 | 47 | | | |
| 37 | 7 | 200 | 0 | 6 | 200 | 46 | | | |
| 38 | 9 | 176 | 0 | 7 | 176 | 63 | | | |
| 39 | 3 | 11 | 0 | 3 | 11 | 78 | | | |
| 40 | 3 | 33 | 0 | 3 | 33 | 62 | | | |
| 41 | 11 | 134 | 0 | 7 | 134 | 62 | | | |
| 42 | 18 | 157 | 0 | 14 | 157 | 47 | | | |
| 43 | 9 | 99 | 0 | 6 | 99 | 62 | | | |
| 44 | 3 | 44 | 0 | 3 | 44 | 47 | | | |
| 45 | 12 | 151 | 0 | 11 | 151 | 62 | | | |
| 46 | 12 | 172 | 0 | 11 | 172 | 47 | | | |
| 47 | 10 | 140 | 0 | 6 | 140 | 46 | | | |
| 48 | 8 | 154 | 0 | 5 | 154 | 47 | | | |
| 49 | 3 | 29 | 0 | 3 | 29 | 62 | | | |
| 50 | 6 | 139 | 0 | 6 | 139 | 63 | | | |
| 51 | 8 | 176 | 0 | 7 | 176 | 62 | | | |
| 52 | 6 | 147 | 0 | 5 | 147 | 78 | | | |
| 53 | 4 | 50 | 0 | 4 | 50 | 62 | | | |
| 54 | 12 | 143 | 0 | 9 | 143 | 63 | | | |
| 55 | 4 | 54 | 0 | 4 | 54 | 62 | | | |
| 56 | 12 | 133 | 1 | 11 | 133 | 47 | | | |
| 57 | 13 | 159 | 0 | 8 | 159 | 63 | | | |
| 58 | 18 | 189 | 0 | 14 | 189 | 62 | | | |
| 59 | 11 | 134 | 0 | 7 | 134 | 62 | | | |
| 60 | 9 | 157 | 0 | 9 | 157 | 63 | | | |
| 61 | 2 | 28 | 0 | 2 | 28 | 62 | | | |
| 62 | 10 | 126 | 0 | 6 | 126 | 47 | | | |
| 63 | 10 | 106 | 0 | 9 | 106 | 46 | | | |
| 64 | 9 | 155 | 0 | 6 | 155 | 63 | | | |
| 65 | 10 | 129 | 0 | 6 | 129 | 62 | | | |
| 66 | 4 | 39 | 0 | 4 | 39 | 47 | | | |
| 67 | 12 | 134 | 0 | 8 | 134 | 62 | | | |
| 68 | 11 | 195 | 0 | 9 | 195 | 47 | | | |
| 69 | 8 | 177 | 0 | 8 | 177 | 47 | | | |
| 70 | 2 | 27 | 0 | 2 | 27 | 62 | | | |
| 71 | 3 | 43 | 0 | 3 | 43 | 47 | | | |
| 72 | 4 | 37 | 0 | 4 | 37 | 62 | | | |
| 73 | 8 | 127 | 0 | 7 | 127 | 47 | | | |
| 74 | 8 | 159 | 0 | 7 | 159 | 62 | | | |
| 75 | 4 | 48 | 0 | 3 | 48 | 63 | | | |
| 76 | 18 | 142 | 0 | 14 | 142 | 62 | | | |
| 77 | 3 | 46 | 0 | 3 | 46 | 63 | | | |
| 78 | 6 | 163 | 0 | 5 | 163 | 46 | | | |
| 79 | 10 | 155 | 0 | 9 | 155 | 63 | | | |
| 80 | 4 | 38 | 0 | 4 | 38 | 47 | | | |
| Total | 654 | 9583 | 5 | 533 | 9583 | 4711 | 99 | 1466 | 12213998 |
| Prom | 8.175 | 119.7875 | 4305.063 | 6.6625 | 119.7875 | 4363.888 | 9.9 | 146.6 | 1221400 |
| ValMin | 2 | 11 | 0 | 2 | 11 | 46 | 3 | 27 | 68 |
| ValMax | 18 | 212 | 2 | 14 | 212 | 125 | 14 | 236 | 5356091 |
| DesvEst | 3.873882 | 55.41208 | 0.331615 | 2.942245 | 55.41208 | 11.89085 | 3.17805 | 69.73474 | 1884209 |

Tabla A.3: Ambiente 2. Resolución: 20cm. Propagación de frente de onda: 2082ms

| Pruebas | SVP | | | SVA | | | MA | | |
|----------------|-------------------|-----------------|----------------|-------------------|-----------------|-----------------|-------------------|-----------------|-----------------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 66 | 320 | 2 | 72 | 320 | 1 | 17 | 320 | 0 |
| 2 | 27 | 211 | 1 | 26 | 211 | 1 | 12 | 211 | 1 |
| 3 | 7 | 53 | 0 | 22 | 53 | 0 | 6 | 53 | 0 |
| 4 | 15 | 83 | 0 | 35 | 83 | 0 | 4 | 83 | 1 |
| 5 | 4 | 50 | 0 | 9 | 50 | 0 | 3 | 50 | 0 |
| 6 | 43 | 272 | 2 | 70 | 272 | 1 | 13 | 272 | 1 |
| 7 | 26 | 231 | 0 | 26 | 231 | 1 | 12 | 231 | 0 |
| 8 | 43 | 261 | 0 | 83 | 261 | 2 | 14 | 261 | 1 |
| 9 | 40 | 205 | 2 | 45 | 205 | 1 | 14 | 205 | 1 |
| 10 | 38 | 228 | 0 | 40 | 228 | 1 | 14 | 228 | 0 |
| 11 | 52 | 247 | 0 | 50 | 247 | 1 | 9 | 247 | 0 |
| 12 | 73 | 317 | 1 | 94 | 317 | 2 | 15 | 317 | 1 |
| 13 | 53 | 240 | 1 | 84 | 240 | 1 | 11 | 240 | 1 |
| 14 | 73 | 291 | 1 | 73 | 291 | 3 | 15 | 291 | 1 |
| 15 | 35 | 317 | 0 | 72 | 317 | 1 | 14 | 317 | 1 |
| 16 | 7 | 55 | 0 | 21 | 55 | 1 | 7 | 55 | 0 |
| 17 | 65 | 227 | 0 | 48 | 227 | 1 | 14 | 227 | 0 |
| 18 | 31 | 236 | 0 | 45 | 236 | 1 | 15 | 236 | 0 |
| 19 | 13 | 44 | 0 | 19 | 44 | 0 | 4 | 44 | 0 |
| 20 | 45 | 217 | 1 | 59 | 217 | 1 | 11 | 217 | 1 |
| 21 | 9 | 62 | 0 | 10 | 62 | 1 | 3 | 62 | 0 |
| 22 | 23 | 215 | 1 | 27 | 215 | 1 | 13 | 215 | 1 |
| 23 | 37 | 239 | 1 | 43 | 239 | 1 | 12 | 239 | 1 |
| 24 | 69 | 269 | 0 | 79 | 269 | 1 | 15 | 269 | 0 |
| 25 | 65 | 282 | 1 | 81 | 282 | 3 | 17 | 282 | 0 |
| 26 | 71 | 280 | 0 | 67 | 280 | 1 | 15 | 280 | 1 |
| 27 | 11 | 43 | 0 | 10 | 43 | 0 | 3 | 43 | 1 |
| 28 | 49 | 230 | 1 | 50 | 230 | 1 | 11 | 230 | 0 |
| 29 | 61 | 270 | 1 | 65 | 270 | 1 | 15 | 270 | 0 |
| 30 | 5 | 47 | 0 | 34 | 47 | 1 | 4 | 47 | 0 |
| 31 | 70 | 314 | 0 | 82 | 314 | 6 | 2 | 314 | 7 |
| 32 | 41 | 251 | 0 | 40 | 251 | 0 | 7 | 251 | 0 |
| 33 | 37 | 249 | 0 | 57 | 249 | 0 | 15 | 249 | 0 |
| 34 | 68 | 262 | 1 | 58 | 262 | 1 | 11 | 262 | 0 |
| 35 | 6 | 78 | 0 | 25 | 78 | 0 | 5 | 78 | 0 |
| 36 | 43 | 249 | 1 | 58 | 249 | 1 | 11 | 249 | 1 |
| 37 | 44 | 243 | 0 | 45 | 243 | 0 | 10 | 243 | 0 |
| 38 | 7 | 74 | 0 | 36 | 74 | 0 | 4 | 74 | 6 |
| 39 | 31 | 245 | 0 | 41 | 245 | 0 | 13 | 245 | 1 |
| 40 | 21 | 229 | 0 | 43 | 229 | 0 | 10 | 229 | 0 |
| 41 | 68 | 331 | 1 | 89 | 331 | 1 | 17 | 331 | 1 |
| 42 | 41 | 259 | 1 | 57 | 259 | 0 | 8 | 259 | 0 |
| 43 | 60 | 256 | 1 | 47 | 256 | 1 | 10 | 256 | 0 |
| 44 | 43 | 330 | 0 | 75 | 330 | 1 | 14 | 330 | 0 |
| 45 | 42 | 246 | 1 | 43 | 246 | 1 | 14 | 246 | 0 |
| 46 | 45 | 226 | 1 | 43 | 226 | 0 | 8 | 226 | 0 |
| 47 | 36 | 279 | 1 | 74 | 279 | 1 | 8 | 279 | 1 |
| 48 | 43 | 335 | 1 | 76 | 335 | 1 | 15 | 335 | 0 |
| 49 | 43 | 257 | 1 | 50 | 257 | 1 | 12 | 257 | 0 |
| 50 | 70 | 250 | 1 | 88 | 250 | 1 | 10 | 250 | 0 |
| 51 | 65 | 237 | 1 | 64 | 237 | 1 | 13 | 237 | 0 |
| 52 | 67 | 258 | 0 | 77 | 258 | 0 | 12 | 258 | 0 |
| 53 | 45 | 270 | 0 | 55 | 270 | 0 | 12 | 270 | 0 |
| 54 | 36 | 232 | 0 | 44 | 232 | 1 | 15 | 232 | 0 |
| 55 | 60 | 318 | 0 | 65 | 318 | 1 | 13 | 318 | 0 |
| 56 | 2 | 20 | 0 | 7 | 20 | 0 | 2 | 20 | 0 |
| 57 | 45 | 237 | 0 | 63 | 237 | 0 | 11 | 237 | 0 |
| 58 | 7 | 90 | 0 | 44 | 90 | 0 | 7 | 90 | 1 |
| 59 | 5 | 29 | 0 | 20 | 29 | 0 | 4 | 29 | 0 |
| 60 | 36 | 236 | 0 | 37 | 236 | 0 | 11 | 236 | 0 |
| 61 | 45 | 247 | 0 | 48 | 247 | 0 | 18 | 247 | 0 |
| 62 | 65 | 247 | 1 | 47 | 247 | 0 | 15 | 247 | 1 |
| 63 | 68 | 307 | 0 | 78 | 307 | 1 | 14 | 307 | 0 |
| 64 | 29 | 236 | 1 | 37 | 236 | 0 | 9 | 236 | 1 |
| 65 | 50 | 215 | 0 | 40 | 215 | 0 | 14 | 215 | 0 |
| 66 | 5 | 57 | 0 | 26 | 57 | 0 | 5 | 57 | 0 |
| 67 | 43 | 343 | 1 | 64 | 343 | 0 | 15 | 343 | 1 |
| 68 | 45 | 264 | 0 | 75 | 264 | 1 | 12 | 264 | 0 |
| 69 | 38 | 264 | 1 | 49 | 264 | 0 | 14 | 264 | 0 |
| 70 | 45 | 237 | 0 | 78 | 237 | 1 | 11 | 237 | 0 |
| 71 | 40 | 223 | 0 | 39 | 223 | 0 | 8 | 223 | 0 |
| 72 | 43 | 345 | 0 | 75 | 345 | 0 | 14 | 345 | 0 |
| 73 | 3 | 17 | 0 | 12 | 17 | 0 | 3 | 17 | 0 |
| 74 | 60 | 251 | 0 | 55 | 251 | 0 | 9 | 251 | 0 |
| 75 | 7 | 68 | 0 | 30 | 68 | 0 | 4 | 68 | 0 |
| 76 | 57 | 238 | 0 | 58 | 238 | 1 | 17 | 238 | 0 |
| 77 | 77 | 324 | 0 | 84 | 324 | 0 | 18 | 324 | 0 |
| 78 | 3 | 36 | 0 | 13 | 36 | 0 | 3 | 36 | 0 |
| 79 | 73 | 316 | 0 | 88 | 316 | 1 | 16 | 316 | 0 |
| 80 | 5 | 53 | 0 | 37 | 53 | 0 | 4 | 53 | 0 |
| Suma | 3184 | 17220 | 32 | 4095 | 17220 | 55 | 859 | 17220 | 35 |
| Prom | 39.8 | 215.25 | 2082.4 | 51.1875 | 215.25 | 2082.688 | 10.7375 | 215.25 | 2082.438 |
| ValMin | 2 | 17 | 0 | 7 | 17 | 0 | 2 | 17 | 0 |
| ValMax | 77 | 345 | 2 | 94 | 345 | 6 | 18 | 345 | 7 |
| DesvEst | 22.13789 | 94.14176 | 0.56479 | 22.46465 | 94.14176 | 0.894339 | 4.479895 | 94.14176 | 1.077312 |

Tabla A.4: Ambiente 2. Resolución: 20cm. Propagación de frente de onda: 2082m/s

| Pruebas | MAVC | | | GR | | | A*RR | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 16 | 320 | 1 | 14 | 320 | 70 | 19 | 398 | 1844880 |
| 2 | 12 | 211 | 0 | 11 | 211 | 83 | 17 | 213 | 13697356 |
| 3 | 6 | 53 | 0 | 6 | 53 | 61 | 6 | 61 | 379086 |
| 4 | 4 | 83 | 0 | 4 | 83 | 60 | 6 | 83 | 304065 |
| 5 | 3 | 50 | 0 | 3 | 50 | 58 | 4 | 51 | 40 |
| 6 | 16 | 272 | 1 | 11 | 272 | 78 | | | |
| 7 | 12 | 231 | 0 | 12 | 231 | 53 | | | |
| 8 | 14 | 261 | 1 | 9 | 261 | 57 | | | |
| 9 | 14 | 205 | 0 | 13 | 205 | 65 | | | |
| 10 | 14 | 228 | 0 | 13 | 228 | 56 | | | |
| 11 | 10 | 247 | 0 | 9 | 247 | 52 | | | |
| 12 | 16 | 317 | 1 | 13 | 317 | 55 | | | |
| 13 | 12 | 240 | 1 | 9 | 240 | 54 | | | |
| 14 | 17 | 291 | 1 | 14 | 291 | 62 | | | |
| 15 | 13 | 317 | 0 | 13 | 317 | 54 | | | |
| 16 | 6 | 55 | 0 | 6 | 55 | 55 | | | |
| 17 | 15 | 227 | 1 | 13 | 227 | 53 | | | |
| 18 | 15 | 236 | 1 | 11 | 236 | 56 | | | |
| 19 | 5 | 44 | 0 | 4 | 44 | 60 | | | |
| 20 | 12 | 217 | 0 | 9 | 217 | 60 | | | |
| 21 | 3 | 62 | 0 | 3 | 62 | 52 | | | |
| 22 | 13 | 215 | 1 | 11 | 215 | 67 | | | |
| 23 | 16 | 239 | 1 | 11 | 239 | 67 | | | |
| 24 | 14 | 269 | 1 | 13 | 269 | 57 | | | |
| 25 | 14 | 282 | 1 | 14 | 282 | 54 | | | |
| 26 | 14 | 280 | 0 | 13 | 280 | 55 | | | |
| 27 | 3 | 43 | 0 | 3 | 43 | 51 | | | |
| 28 | 11 | 230 | 1 | 8 | 230 | 64 | | | |
| 29 | 14 | 270 | 0 | 13 | 270 | 52 | | | |
| 30 | 4 | 47 | 0 | 4 | 47 | 52 | | | |
| 31 | 2 | 314 | 2 | 2 | 314 | 62 | | | |
| 32 | 12 | 251 | 0 | 7 | 251 | 53 | | | |
| 33 | 16 | 249 | 0 | 11 | 249 | 52 | | | |
| 34 | 14 | 262 | 0 | 11 | 262 | 54 | | | |
| 35 | 5 | 78 | 0 | 5 | 78 | 53 | | | |
| 36 | 16 | 249 | 0 | 11 | 249 | 52 | | | |
| 37 | 10 | 243 | 0 | 9 | 243 | 52 | | | |
| 38 | 4 | 74 | 1 | 4 | 74 | 62 | | | |
| 39 | 13 | 245 | 1 | 11 | 245 | 53 | | | |
| 40 | 10 | 229 | 0 | 10 | 229 | 59 | | | |
| 41 | 16 | 331 | 0 | 14 | 331 | 53 | | | |
| 42 | 13 | 259 | 1 | 8 | 259 | 62 | | | |
| 43 | 11 | 256 | 0 | 9 | 256 | 53 | | | |
| 44 | 14 | 330 | 0 | 13 | 330 | 53 | | | |
| 45 | 14 | 246 | 0 | 13 | 246 | 52 | | | |
| 46 | 9 | 226 | 0 | 7 | 226 | 52 | | | |
| 47 | 9 | 279 | 0 | 8 | 279 | 53 | | | |
| 48 | 15 | 335 | 0 | 14 | 335 | 57 | | | |
| 49 | 16 | 257 | 0 | 11 | 257 | 52 | | | |
| 50 | 14 | 250 | 0 | 10 | 250 | 62 | | | |
| 51 | 14 | 237 | 0 | 13 | 237 | 56 | | | |
| 52 | 13 | 258 | 0 | 10 | 258 | 53 | | | |
| 53 | 16 | 270 | 0 | 11 | 270 | 56 | | | |
| 54 | 16 | 232 | 0 | 13 | 232 | 52 | | | |
| 55 | 13 | 318 | 0 | 11 | 318 | 59 | | | |
| 56 | 2 | 20 | 0 | 2 | 20 | 54 | | | |
| 57 | 12 | 237 | 0 | 10 | 237 | 56 | | | |
| 58 | 6 | 90 | 0 | 6 | 90 | 54 | | | |
| 59 | 4 | 29 | 0 | 4 | 29 | 55 | | | |
| 60 | 12 | 236 | 1 | 10 | 236 | 52 | | | |
| 61 | 14 | 247 | 0 | 13 | 247 | 54 | | | |
| 62 | 15 | 247 | 0 | 14 | 247 | 55 | | | |
| 63 | 14 | 307 | 1 | 11 | 307 | 52 | | | |
| 64 | 9 | 236 | 0 | 7 | 236 | 52 | | | |
| 65 | 16 | 215 | 0 | 14 | 215 | 53 | | | |
| 66 | 5 | 57 | 0 | 4 | 57 | 53 | | | |
| 67 | 15 | 343 | 0 | 14 | 343 | 55 | | | |
| 68 | 16 | 264 | 0 | 12 | 264 | 54 | | | |
| 69 | 14 | 264 | 0 | 14 | 264 | 54 | | | |
| 70 | 12 | 237 | 0 | 9 | 237 | 53 | | | |
| 71 | 8 | 223 | 0 | 7 | 223 | 54 | | | |
| 72 | 14 | 345 | 0 | 14 | 345 | 53 | | | |
| 73 | 3 | 17 | 0 | 3 | 17 | 52 | | | |
| 74 | 11 | 251 | 0 | 8 | 251 | 57 | | | |
| 75 | 4 | 68 | 0 | 4 | 68 | 52 | | | |
| 76 | 15 | 238 | 0 | 13 | 238 | 53 | | | |
| 77 | 18 | 324 | 1 | 16 | 324 | 52 | | | |
| 78 | 3 | 36 | 0 | 3 | 36 | 57 | | | |
| 79 | 16 | 316 | 1 | 13 | 316 | 52 | | | |
| 80 | 4 | 53 | 0 | 4 | 53 | 52 | | | |
| Suma | 905 | 17220 | 22 | 765 | 17220 | 4490 | 52 | 806 | 16225427 |
| Prom | 11.3125 | 215.25 | 2082.275 | 9.5625 | 215.25 | 2138.125 | 10.4 | 161.2 | 3245085 |
| ValMin | 2 | 14 | 0 | 2 | 17 | 51 | 4 | 51 | 40 |
| ValMax | 18 | 345 | 2 | 16 | 345 | 83 | 19 | 398 | 13697356 |
| DesvEst | 4.580072 | 94.14176 | 0.476671 | 3.781513 | 94.14176 | 5.719996 | 7.021396 | 147.5269 | 5886519 |

Tabla A.5: Ambiente 3. Resolución: 20cm. Propagación de frente de onda: 831ms

| Prueba | SVP | | | SVA | | | MA | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 32 | 270 | 2 | 44 | 270 | 1 | 9 | 270 | 0 |
| 2 | 32 | 242 | 1 | 33 | 242 | 1 | 9 | 242 | 0 |
| 3 | 57 | 212 | 1 | 59 | 212 | 1 | 8 | 212 | 1 |
| 4 | 14 | 194 | 0 | 29 | 194 | 1 | 6 | 194 | 1 |
| 5 | 205 | 823 | 8 | 231 | 823 | 3 | 47 | 823 | 1 |
| 6 | 207 | 834 | 2 | 243 | 834 | 3 | 50 | 834 | 1 |
| 7 | 91 | 284 | 0 | 69 | 284 | 1 | 11 | 284 | 5 |
| 8 | 224 | 824 | 6 | 239 | 824 | 8 | 50 | 824 | 4 |
| 9 | 201 | 731 | 6 | 218 | 731 | 5 | 50 | 731 | 1 |
| 10 | 234 | 822 | 2 | 260 | 822 | 9 | 48 | 822 | 2 |
| 11 | 32 | 293 | 1 | 38 | 293 | 1 | 8 | 293 | 0 |
| 12 | 20 | 189 | 3 | 35 | 189 | 11 | 6 | 189 | 1 |
| 13 | 52 | 215 | 0 | 61 | 215 | 0 | 7 | 215 | 0 |
| 14 | 126 | 534 | 1 | 148 | 534 | 13 | 35 | 534 | 2 |
| 15 | 153 | 603 | 5 | 162 | 603 | 0 | 32 | 603 | 0 |
| 16 | 35 | 184 | 0 | 32 | 184 | 0 | 10 | 184 | 0 |
| 17 | 136 | 541 | 0 | 151 | 541 | 0 | 38 | 541 | 3 |
| 18 | 164 | 612 | 0 | 134 | 612 | 0 | 26 | 612 | 0 |
| 19 | 230 | 847 | 0 | 255 | 847 | 0 | 59 | 847 | 0 |
| 20 | 152 | 595 | 0 | 129 | 595 | 0 | 27 | 595 | 0 |
| 21 | 32 | 187 | 0 | 36 | 187 | 1 | 9 | 187 | 0 |
| 22 | 6 | 156 | 0 | 10 | 156 | 0 | 6 | 156 | 0 |
| 23 | 19 | 195 | 1 | 30 | 195 | 1 | 9 | 195 | 11 |
| 24 | 20 | 183 | 0 | 38 | 183 | 0 | 6 | 183 | 0 |
| 25 | 110 | 303 | 0 | 95 | 303 | 1 | 11 | 303 | 0 |
| 26 | 32 | 266 | 0 | 57 | 266 | 0 | 8 | 266 | 0 |
| 27 | 223 | 854 | 0 | 262 | 854 | 0 | 49 | 854 | 0 |
| 28 | 119 | 506 | 0 | 129 | 506 | 0 | 26 | 506 | 2 |
| 29 | 22 | 175 | 0 | 37 | 175 | 0 | 5 | 175 | 0 |
| 30 | 34 | 239 | 0 | 60 | 239 | 2 | 8 | 239 | 0 |
| 31 | 221 | 842 | 0 | 248 | 842 | 0 | 48 | 842 | 3 |
| 32 | 32 | 311 | 1 | 35 | 311 | 0 | 8 | 311 | 5 |
| 33 | 10 | 118 | 0 | 18 | 118 | 0 | 5 | 118 | 0 |
| 34 | 1 | 2 | 0 | 2 | 2 | 0 | 1 | 2 | 0 |
| 35 | 228 | 814 | 0 | 253 | 814 | 5 | 47 | 814 | 0 |
| 36 | 119 | 513 | 1 | 131 | 513 | 0 | 26 | 513 | 0 |
| 37 | 189 | 715 | 0 | 205 | 715 | 0 | 50 | 715 | 4 |
| 38 | 133 | 524 | 0 | 151 | 524 | 4 | 33 | 524 | 0 |
| 39 | 231 | 839 | 0 | 247 | 839 | 0 | 49 | 839 | 0 |
| 40 | 230 | 831 | 5 | 282 | 831 | 0 | 59 | 831 | 0 |
| 41 | 18 | 194 | 0 | 20 | 194 | 0 | 9 | 194 | 2 |
| 42 | 32 | 186 | 1 | 44 | 186 | 0 | 9 | 186 | 0 |
| 43 | 230 | 833 | 0 | 267 | 833 | 0 | 62 | 833 | 0 |
| 44 | 32 | 195 | 0 | 38 | 195 | 0 | 8 | 195 | 2 |
| 45 | 250 | 844 | 1 | 272 | 844 | 5 | 5 | 844 | 3 |
| 46 | 147 | 603 | 0 | 149 | 603 | 0 | 31 | 603 | 0 |
| 47 | 255 | 860 | 0 | 268 | 860 | 0 | 51 | 860 | 0 |
| 48 | 185 | 736 | 0 | 225 | 736 | 0 | 44 | 736 | 1 |
| 49 | 23 | 281 | 0 | 59 | 281 | 0 | 14 | 281 | 1 |
| 50 | 126 | 529 | 0 | 134 | 529 | 2 | 36 | 529 | 2 |
| 51 | 250 | 842 | 0 | 248 | 842 | 0 | 6 | 842 | 0 |
| 52 | 132 | 511 | 1 | 125 | 511 | 0 | 26 | 511 | 0 |
| 53 | 3 | 59 | 1 | 6 | 59 | 1 | 3 | 59 | 2 |
| 54 | 32 | 268 | 1 | 39 | 268 | 0 | 9 | 268 | 1 |
| 55 | 157 | 658 | 1 | 159 | 658 | 3 | 34 | 658 | 0 |
| 56 | 178 | 724 | 1 | 189 | 724 | 0 | 41 | 724 | 3 |
| 57 | 205 | 825 | 1 | 239 | 825 | 0 | 48 | 825 | 2 |
| 58 | 19 | 205 | 0 | 33 | 205 | 1 | 9 | 205 | 0 |
| 59 | 13 | 202 | 1 | 18 | 202 | 0 | 7 | 202 | 0 |
| 60 | 15 | 206 | 0 | 29 | 206 | 0 | 12 | 206 | 0 |
| 61 | 137 | 529 | 1 | 152 | 529 | 0 | 25 | 529 | 1 |
| 62 | 32 | 192 | 0 | 28 | 192 | 0 | 10 | 192 | 0 |
| 63 | 4 | 11 | 0 | 5 | 11 | 0 | 3 | 11 | 0 |
| 64 | 46 | 200 | 0 | 44 | 200 | 0 | 10 | 200 | 2 |
| 65 | 32 | 285 | 0 | 39 | 285 | 1 | 8 | 285 | 0 |
| 66 | 5 | 56 | 0 | 15 | 56 | 0 | 4 | 56 | 6 |
| 67 | 131 | 621 | 0 | 153 | 621 | 4 | 31 | 621 | 0 |
| 68 | 32 | 238 | 0 | 44 | 238 | 5 | 9 | 238 | 0 |
| 69 | 228 | 812 | 0 | 242 | 812 | 0 | 47 | 812 | 5 |
| 70 | 168 | 624 | 1 | 165 | 624 | 1 | 28 | 624 | 0 |
| 71 | 3 | 18 | 1 | 11 | 18 | 1 | 3 | 18 | 6 |
| 72 | 140 | 616 | 1 | 137 | 616 | 0 | 32 | 616 | 0 |
| 73 | 215 | 782 | 0 | 247 | 782 | 0 | 52 | 782 | 0 |
| 74 | 157 | 608 | 0 | 153 | 608 | 0 | 30 | 608 | 0 |
| 75 | 158 | 604 | 0 | 144 | 604 | 0 | 26 | 604 | 0 |
| 76 | 250 | 837 | 6 | 244 | 837 | 9 | 4 | 837 | 5 |
| 77 | 181 | 689 | 0 | 190 | 689 | 0 | 37 | 689 | 0 |
| 78 | 11 | 165 | 0 | 15 | 165 | 1 | 5 | 165 | 0 |
| 79 | 238 | 825 | 0 | 247 | 825 | 0 | 48 | 825 | 2 |
| 80 | 136 | 628 | 0 | 157 | 628 | 0 | 33 | 628 | 0 |
| Suma | 9014 | 37523 | 65 | 9889 | 37523 | 106 | 1908 | 37523 | 93 |
| Prom | 112.675 | 469.0375 | 831.8125 | 123.6125 | 469.0375 | 832.325 | 23.85 | 469.0375 | 832.1625 |
| ValMin | 1 | 2 | 0 | 2 | 2 | 0 | 1 | 2 | 0 |
| ValMax | 255 | 860 | 8 | 280 | 860 | 13 | 62 | 860 | 11 |
| DesvEst | 86.64357 | 277.9272 | 1.631135 | 91.86486 | 277.9272 | 2.656613 | 18.28751 | 277.9272 | 1.958065 |

Tabla A.6: Ambiente 3. Resolución: 20cm. Propagación de frente de onda: 831ms

| Prueba | MAVC | | | GR | | | A*RR | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 9 | 270 | 0 | 8 | 270 | 63 | 10 | 274 | 228959 |
| 2 | 9 | 242 | 1 | 8 | 242 | 59 | 10 | 244 | 220314 |
| 3 | 7 | 212 | 0 | 7 | 212 | 65 | 9 | 212 | 2758 |
| 4 | 6 | 194 | 0 | 6 | 194 | 53 | 7 | 194 | 114 |
| 5 | 48 | 823 | 1 | 38 | 823 | 83 | 47 | 839 | 7344552 |
| 6 | 50 | 834 | 1 | 40 | 834 | 76 | 45 | 844 | 7415433 |
| 7 | 11 | 284 | 1 | 11 | 284 | 94 | 14 | 286 | 110858 |
| 8 | 51 | 824 | 2 | 39 | 824 | 112 | 50 | 842 | 7238595 |
| 9 | 51 | 731 | 1 | 39 | 731 | 117 | 49 | 747 | 7426063 |
| 10 | 50 | 822 | 9 | 38 | 822 | 175 | 48 | 838 | 7340040 |
| 11 | 8 | 293 | 1 | 8 | 293 | 60 | 10 | 295 | 237034 |
| 12 | 6 | 189 | 0 | 6 | 189 | 87 | 7 | 189 | 153 |
| 13 | 7 | 215 | 0 | 7 | 215 | 61 | 10 | 219 | 1193 |
| 14 | 36 | 534 | 0 | 30 | 534 | 56 | 28 | 538 | 2581133 |
| 15 | 33 | 603 | 0 | 24 | 603 | 78 | | | |
| 16 | 9 | 184 | 2 | 7 | 184 | 78 | | | |
| 17 | 38 | 541 | 0 | 32 | 541 | 47 | | | |
| 18 | 29 | 612 | 0 | 25 | 612 | 63 | | | |
| 19 | 59 | 847 | 0 | 43 | 847 | 2293 | | | |
| 20 | 28 | 595 | 0 | 24 | 595 | 63 | | | |
| 21 | 9 | 187 | 15 | 8 | 187 | 63 | | | |
| 22 | 5 | 156 | 0 | 5 | 156 | 46 | | | |
| 23 | 9 | 195 | 10 | 8 | 195 | 63 | | | |
| 24 | 6 | 183 | 0 | 6 | 183 | 62 | | | |
| 25 | 13 | 303 | 0 | 11 | 303 | 47 | | | |
| 26 | 8 | 266 | 0 | 8 | 266 | 47 | | | |
| 27 | 52 | 854 | 0 | 40 | 854 | 124 | | | |
| 28 | 27 | 506 | 1 | 21 | 506 | 47 | | | |
| 29 | 5 | 175 | 0 | 5 | 175 | 62 | | | |
| 30 | 8 | 239 | 0 | 8 | 239 | 47 | | | |
| 31 | 51 | 842 | 0 | 39 | 842 | 110 | | | |
| 32 | 8 | 311 | 4 | 8 | 311 | 63 | | | |
| 33 | 4 | 118 | 0 | 4 | 118 | 46 | | | |
| 34 | 1 | 2 | 0 | 1 | 2 | 47 | | | |
| 35 | 50 | 814 | 3 | 38 | 814 | 63 | | | |
| 36 | 27 | 513 | 0 | 21 | 513 | 62 | | | |
| 37 | 51 | 715 | 5 | 38 | 715 | 78 | | | |
| 38 | 37 | 524 | 0 | 27 | 524 | 47 | | | |
| 39 | 52 | 839 | 0 | 39 | 839 | 171 | | | |
| 40 | 59 | 831 | 0 | 46 | 831 | 3307 | | | |
| 41 | 9 | 194 | 9 | 8 | 194 | 47 | | | |
| 42 | 9 | 186 | 0 | 8 | 186 | 62 | | | |
| 43 | 61 | 833 | 0 | 48 | 833 | 4306 | | | |
| 44 | 8 | 195 | 2 | 8 | 195 | 62 | | | |
| 45 | 5 | 844 | 0 | 5 | 844 | 78 | | | |
| 46 | 32 | 603 | 0 | 25 | 603 | 47 | | | |
| 47 | 53 | 860 | 3 | 41 | 860 | 78 | | | |
| 48 | 45 | 736 | 0 | 34 | 736 | 63 | | | |
| 49 | 15 | 281 | 0 | 12 | 281 | 63 | | | |
| 50 | 36 | 529 | 0 | 30 | 529 | 63 | | | |
| 51 | 6 | 842 | 2 | 6 | 842 | 63 | | | |
| 52 | 27 | 511 | 0 | 22 | 511 | 47 | | | |
| 53 | 3 | 59 | 0 | 3 | 59 | 63 | | | |
| 54 | 9 | 268 | 0 | 8 | 268 | 47 | | | |
| 55 | 35 | 658 | 1 | 27 | 658 | 47 | | | |
| 56 | 42 | 724 | 0 | 31 | 724 | 63 | | | |
| 57 | 49 | 825 | 0 | 39 | 825 | 78 | | | |
| 58 | 9 | 205 | 0 | 8 | 205 | 47 | | | |
| 59 | 7 | 202 | 3 | 7 | 202 | 63 | | | |
| 60 | 12 | 206 | 0 | 11 | 206 | 47 | | | |
| 61 | 26 | 529 | 2 | 24 | 529 | 47 | | | |
| 62 | 9 | 192 | 0 | 8 | 192 | 47 | | | |
| 63 | 3 | 11 | 0 | 3 | 11 | 46 | | | |
| 64 | 9 | 200 | 4 | 7 | 200 | 47 | | | |
| 65 | 8 | 285 | 0 | 8 | 285 | 46 | | | |
| 66 | 4 | 56 | 0 | 4 | 56 | 47 | | | |
| 67 | 32 | 621 | 0 | 25 | 621 | 47 | | | |
| 68 | 9 | 238 | 0 | 8 | 238 | 47 | | | |
| 69 | 50 | 812 | 7 | 38 | 812 | 63 | | | |
| 70 | 31 | 624 | 0 | 27 | 624 | 63 | | | |
| 71 | 4 | 18 | 6 | 3 | 18 | 63 | | | |
| 72 | 34 | 616 | 0 | 27 | 616 | 62 | | | |
| 73 | 53 | 782 | 1 | 41 | 782 | 312 | | | |
| 74 | 31 | 608 | 0 | 24 | 608 | 62 | | | |
| 75 | 29 | 604 | 5 | 25 | 604 | 62 | | | |
| 76 | 4 | 837 | 0 | 4 | 837 | 62 | | | |
| 77 | 36 | 689 | 0 | 29 | 689 | 62 | | | |
| 78 | 5 | 165 | 0 | 5 | 165 | 47 | | | |
| 79 | 50 | 825 | 1 | 38 | 825 | 78 | | | |
| 80 | 34 | 628 | 0 | 27 | 628 | 62 | | | |
| Suma | 1960 | 37523 | 103 | 1577 | 37523 | 15191 | 344 | 6561 | 40147199 |
| Prom | 24.5 | 469.0375 | 832.2875 | 19.7125 | 469.0375 | 1020.888 | 24.57143 | 468.6429 | 2867657 |
| ValMin | 1 | 2 | 0 | 1 | 2 | 46 | 7 | 189 | 114 |
| ValMax | 61 | 860 | 15 | 48 | 860 | 4306 | 50 | 844 | 7426063 |
| DesvEst | 18.89411 | 277.9272 | 2.705801 | 14.10018 | 277.9272 | 639.6133 | 18.69889 | 286.8727 | 3530681 |

Tabla A.7: Ambiente 4. Resolución: 20cm. Propagación de frente de onda: 2996ms

| Prueba | SVP | | | SVA | | | MA | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 2 | 29 | 0 | 11 | 29 | 0 | 2 | 29 | 0 |
| 2 | 10 | 121 | 0 | 51 | 121 | 0 | 4 | 121 | 0 |
| 3 | 56 | 238 | 0 | 88 | 238 | 0 | 17 | 238 | 0 |
| 4 | 4 | 44 | 0 | 28 | 44 | 0 | 4 | 44 | 0 |
| 5 | 3 | 92 | 0 | 15 | 92 | 0 | 4 | 92 | 0 |
| 6 | 8 | 92 | 0 | 22 | 92 | 0 | 8 | 92 | 0 |
| 7 | 24 | 137 | 1 | 43 | 137 | 0 | 8 | 137 | 0 |
| 8 | 5 | 59 | 0 | 33 | 59 | 0 | 3 | 59 | 0 |
| 9 | 15 | 119 | 0 | 35 | 119 | 0 | 6 | 119 | 0 |
| 10 | 45 | 203 | 0 | 59 | 203 | 0 | 8 | 203 | 0 |
| 11 | 46 | 182 | 0 | 51 | 182 | 0 | 6 | 182 | 0 |
| 12 | 29 | 76 | 0 | 24 | 76 | 0 | 3 | 76 | 0 |
| 13 | 42 | 214 | 0 | 78 | 214 | 0 | 6 | 214 | 0 |
| 14 | 15 | 165 | 0 | 29 | 165 | 0 | 7 | 165 | 0 |
| 15 | 13 | 115 | 0 | 33 | 115 | 1 | 12 | 115 | 0 |
| 16 | 10 | 116 | 0 | 18 | 116 | 1 | 6 | 116 | 0 |
| 17 | 2 | 43 | 0 | 5 | 43 | 0 | 2 | 43 | 0 |
| 18 | 3 | 37 | 0 | 21 | 37 | 1 | 3 | 37 | 0 |
| 19 | 5 | 121 | 0 | 47 | 121 | 1 | 4 | 121 | 0 |
| 20 | 26 | 185 | 0 | 41 | 185 | 0 | 5 | 185 | 0 |
| 21 | 26 | 132 | 1 | 61 | 132 | 0 | 5 | 132 | 0 |
| 22 | 2 | 11 | 1 | 3 | 11 | 0 | 2 | 11 | 0 |
| 23 | 31 | 135 | 0 | 60 | 135 | 0 | 6 | 135 | 1 |
| 24 | 20 | 239 | 1 | 52 | 239 | 1 | 11 | 239 | 0 |
| 25 | 3 | 47 | 1 | 17 | 47 | 0 | 3 | 47 | 0 |
| 26 | 36 | 216 | 0 | 72 | 216 | 1 | 9 | 216 | 1 |
| 27 | 5 | 58 | 0 | 40 | 58 | 1 | 3 | 58 | 0 |
| 28 | 3 | 55 | 1 | 20 | 55 | 0 | 3 | 55 | 0 |
| 29 | 36 | 200 | 1 | 87 | 200 | 1 | 7 | 200 | 1 |
| 30 | 22 | 180 | 0 | 46 | 180 | 1 | 6 | 180 | 0 |
| 31 | 4 | 61 | 0 | 30 | 61 | 1 | 4 | 61 | 0 |
| 32 | 19 | 62 | 1 | 31 | 62 | 0 | 4 | 62 | 0 |
| 33 | 3 | 59 | 0 | 35 | 59 | 0 | 4 | 59 | 0 |
| 34 | 11 | 146 | 0 | 52 | 146 | 1 | 7 | 146 | 0 |
| 35 | 15 | 171 | 0 | 23 | 171 | 1 | 7 | 171 | 0 |
| 36 | 35 | 115 | 0 | 51 | 115 | 1 | 5 | 115 | 0 |
| 37 | 34 | 213 | 0 | 82 | 213 | 1 | 8 | 213 | 1 |
| 38 | 15 | 173 | 0 | 39 | 173 | 1 | 7 | 173 | 0 |
| 39 | 42 | 214 | 0 | 85 | 214 | 0 | 5 | 214 | 0 |
| 40 | 18 | 218 | 0 | 32 | 218 | 1 | 10 | 218 | 1 |
| 41 | 51 | 188 | 0 | 75 | 188 | 0 | 19 | 188 | 0 |
| 42 | 16 | 180 | 0 | 53 | 180 | 0 | 8 | 180 | 0 |
| 43 | 3 | 56 | 0 | 17 | 56 | 0 | 3 | 56 | 1 |
| 44 | 14 | 232 | 0 | 56 | 232 | 0 | 10 | 232 | 0 |
| 45 | 9 | 45 | 0 | 22 | 45 | 0 | 3 | 45 | 1 |
| 46 | 30 | 241 | 0 | 43 | 241 | 0 | 6 | 241 | 0 |
| 47 | 4 | 41 | 0 | 24 | 41 | 0 | 4 | 41 | 0 |
| 48 | 13 | 114 | 1 | 46 | 114 | 0 | 10 | 114 | 0 |
| 49 | 4 | 43 | 0 | 15 | 43 | 0 | 3 | 43 | 0 |
| 50 | 24 | 114 | 1 | 75 | 114 | 0 | 6 | 114 | 0 |
| 51 | 29 | 201 | 0 | 36 | 201 | 1 | 13 | 201 | 0 |
| 52 | 3 | 31 | 0 | 14 | 31 | 0 | 3 | 31 | 1 |
| 53 | 62 | 193 | 1 | 81 | 193 | 0 | 17 | 193 | 0 |
| 54 | 3 | 38 | 0 | 7 | 38 | 0 | 3 | 38 | 0 |
| 55 | 26 | 231 | 0 | 46 | 231 | 0 | 7 | 231 | 0 |
| 56 | 16 | 188 | 0 | 52 | 188 | 1 | 7 | 188 | 0 |
| 57 | 14 | 163 | 0 | 63 | 163 | 1 | 6 | 163 | 0 |
| 58 | 3 | 57 | 0 | 32 | 57 | 1 | 3 | 57 | 1 |
| 59 | 13 | 132 | 1 | 34 | 132 | 0 | 11 | 132 | 1 |
| 60 | 31 | 254 | 0 | 84 | 254 | 1 | 8 | 254 | 0 |
| 61 | 44 | 125 | 0 | 35 | 125 | 0 | 7 | 125 | 0 |
| 62 | 24 | 162 | 0 | 37 | 162 | 1 | 5 | 162 | 0 |
| 63 | 24 | 232 | 1 | 39 | 232 | 1 | 12 | 232 | 1 |
| 64 | 6 | 66 | 0 | 41 | 66 | 0 | 5 | 66 | 1 |
| 65 | 46 | 217 | 0 | 78 | 217 | 1 | 7 | 217 | 0 |
| 66 | 4 | 57 | 0 | 29 | 57 | 0 | 4 | 57 | 0 |
| 67 | 45 | 102 | 0 | 39 | 102 | 0 | 3 | 102 | 0 |
| 68 | 28 | 103 | 0 | 21 | 103 | 1 | 8 | 103 | 0 |
| 69 | 36 | 201 | 1 | 44 | 201 | 1 | 12 | 201 | 0 |
| 70 | 40 | 117 | 0 | 37 | 117 | 0 | 7 | 117 | 0 |
| 71 | 43 | 258 | 1 | 108 | 258 | 2 | 7 | 258 | 1 |
| 72 | 4 | 49 | 0 | 19 | 49 | 0 | 3 | 49 | 0 |
| 73 | 28 | 223 | 1 | 39 | 223 | 1 | 7 | 223 | 0 |
| 74 | 8 | 109 | 1 | 29 | 109 | 0 | 4 | 109 | 1 |
| 75 | 25 | 162 | 0 | 30 | 162 | 1 | 7 | 162 | 0 |
| 76 | 6 | 33 | 0 | 17 | 33 | 0 | 4 | 33 | 0 |
| 77 | 45 | 196 | 1 | 48 | 196 | 1 | 6 | 196 | 0 |
| 78 | 19 | 112 | 1 | 31 | 112 | 0 | 7 | 112 | 1 |
| 79 | 3 | 66 | 0 | 22 | 66 | 1 | 3 | 66 | 1 |
| 80 | 1 | 11 | 0 | 2 | 11 | 0 | 1 | 11 | 0 |
| Suma | 1590 | 10436 | 19 | 3270 | 10436 | 32 | 503 | 10436 | 16 |
| Prom | 19.875 | 130.45 | 2996.238 | 40.875 | 130.45 | 2996.4 | 6.2875 | 130.45 | 2996.2 |
| ValMin | 1 | 11 | 2996 | 2 | 11 | 2996 | 1 | 11 | 2996 |
| ValMax | 62 | 258 | 2997 | 108 | 258 | 2997 | 19 | 258 | 2997 |
| DesvEst | 15.75716 | 70.76399 | 0.428236 | 22.66899 | 70.76399 | 0.518029 | 3.501153 | 70.76399 | 0.402524 |

Tabla A.8: Ambiente 4. Resolución: 20cm. Propagación de frente de onda: 2996ms

| Prueba | MAVC | | | GR | | | A*RR | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 3 | 29 | 0 | 2 | 29 | 63 | 7 | 87 | 297 |
| 2 | 4 | 121 | 0 | 4 | 121 | 62 | 5 | 121 | 78 |
| 3 | 18 | 238 | 0 | 16 | 238 | 62 | 12 | 256 | 11232 |
| 4 | 4 | 44 | 0 | 4 | 44 | 47 | 5 | 44 | 16 |
| 5 | 4 | 92 | 0 | 3 | 92 | 62 | 6 | 122 | 109 |
| 6 | 7 | 92 | 0 | 7 | 92 | 46 | 9 | 92 | 499 |
| 7 | 8 | 137 | 0 | 7 | 137 | 47 | 8 | 157 | 1310 |
| 8 | 3 | 59 | 0 | 3 | 59 | 47 | 6 | 75 | 78 |
| 9 | 7 | 119 | 0 | 6 | 119 | 47 | 9 | 153 | 42463 |
| 10 | 8 | 203 | 0 | 8 | 203 | 62 | 11 | 207 | 374 |
| 11 | 6 | 182 | 0 | 6 | 182 | 47 | 9 | 196 | 80667 |
| 12 | 3 | 76 | 0 | 3 | 76 | 47 | 8 | 140 | 265 |
| 13 | 6 | 214 | 1 | 6 | 214 | 109 | 11 | 234 | 515 |
| 14 | 7 | 165 | 0 | 6 | 165 | 62 | 7 | 169 | 113888 |
| 15 | 12 | 115 | 1 | 10 | 115 | 54 | 9 | 151 | 515 |
| 16 | 6 | 116 | 0 | 6 | 116 | 61 | 8 | 120 | 325 |
| 17 | 3 | 43 | 0 | 2 | 43 | 54 | 7 | 113 | 205 |
| 18 | 3 | 37 | 0 | 3 | 37 | 61 | 5 | 87 | 57 |
| 19 | 5 | 121 | 1 | 4 | 121 | 90 | 8 | 163 | 199 |
| 20 | 5 | 185 | 0 | 5 | 185 | 68 | 8 | 245 | 110112 |
| 21 | 5 | 132 | 0 | 5 | 132 | 57 | 8 | 150 | 173 |
| 22 | 2 | 11 | 0 | 2 | 11 | 56 | 2 | 11 | 1 |
| 23 | 6 | 135 | 0 | 6 | 135 | 55 | 9 | 145 | 235 |
| 24 | 9 | 239 | 1 | 9 | 239 | 55 | 11 | 261 | 18333 |
| 25 | 3 | 47 | 0 | 3 | 47 | 62 | 8 | 109 | 269 |
| 26 | 8 | 216 | 0 | 7 | 216 | 71 | 13 | 236 | 5210 |
| 27 | 3 | 58 | 0 | 3 | 58 | 55 | 6 | 76 | 58 |
| 28 | 3 | 55 | 0 | 3 | 55 | 55 | 6 | 99 | 77 |
| 29 | 7 | 200 | 0 | 7 | 200 | 55 | 11 | 220 | 4447 |
| 30 | 5 | 180 | 0 | 5 | 180 | 115 | 9 | 182 | 27301 |
| 31 | 4 | 61 | 0 | 4 | 61 | 61 | 5 | 61 | 20 |
| 32 | 3 | 62 | 0 | 3 | 62 | 63 | 6 | 112 | 49 |
| 33 | 3 | 59 | 1 | 3 | 59 | 53 | 6 | 85 | 38 |
| 34 | 7 | 146 | 1 | 5 | 146 | 57 | 11 | 188 | 709 |
| 35 | 7 | 171 | 0 | 6 | 171 | 65 | 7 | 173 | 75073 |
| 36 | 6 | 115 | 1 | 5 | 115 | 58 | 6 | 117 | 28319 |
| 37 | 9 | 213 | 0 | 7 | 213 | 63 | 11 | 249 | 8736 |
| 38 | 8 | 173 | 1 | 6 | 173 | 58 | 8 | 175 | 92255 |
| 39 | 7 | 214 | 0 | 5 | 214 | 61 | 9 | 292 | 146518 |
| 40 | 9 | 218 | 0 | 8 | 218 | 63 | 13 | 264 | 18589 |
| 41 | 18 | 188 | 0 | 16 | 188 | 78 | 13 | 202 | 7378 |
| 42 | 9 | 180 | 0 | 7 | 180 | 63 | 11 | 184 | 483 |
| 43 | 3 | 56 | 0 | 3 | 56 | 47 | 8 | 128 | 1046 |
| 44 | 11 | 232 | 0 | 8 | 232 | 62 | 10 | 294 | 123285 |
| 45 | 4 | 45 | 0 | 3 | 45 | 67 | 8 | 111 | 204 |
| 46 | 6 | 241 | 0 | 5 | 241 | 62 | 9 | 245 | 127121 |
| 47 | 4 | 41 | 0 | 4 | 41 | 60 | 4 | 41 | 12 |
| 48 | 11 | 114 | 0 | 8 | 114 | 61 | 10 | 118 | 6692 |
| 49 | 3 | 43 | 0 | 3 | 43 | 54 | 4 | 61 | 16 |
| 50 | 7 | 114 | 0 | 5 | 114 | 54 | 7 | 114 | 63 |
| 51 | 13 | 201 | 1 | 11 | 201 | 72 | 13 | 257 | 988 |
| 52 | 3 | 31 | 0 | 3 | 31 | 55 | 7 | 97 | 275 |
| 53 | 17 | 193 | 1 | 16 | 193 | 65 | 12 | 211 | 12109 |
| 54 | 3 | 38 | 1 | 3 | 38 | 63 | 3 | 38 | 8 |
| 55 | 6 | 231 | 0 | 6 | 231 | 54 | 13 | 239 | 36687 |
| 56 | 7 | 188 | 0 | 7 | 188 | 57 | 9 | 198 | 12147 |
| 57 | 7 | 163 | 1 | 6 | 163 | 58 | 7 | 165 | 161 |
| 58 | 3 | 57 | 0 | 3 | 57 | 57 | 6 | 87 | 86 |
| 59 | 12 | 132 | 0 | 10 | 132 | 54 | 11 | 180 | 3731 |
| 60 | 7 | 254 | 1 | 7 | 254 | 66 | 11 | 318 | 148510 |
| 61 | 8 | 125 | 0 | 7 | 125 | 55 | 8 | 127 | 285 |
| 62 | 5 | 162 | 1 | 5 | 162 | 54 | 9 | 222 | 140992 |
| 63 | 10 | 232 | 0 | 9 | 232 | 63 | 12 | 270 | 16159 |
| 64 | 6 | 66 | 0 | 5 | 66 | 56 | 6 | 66 | 17 |
| 65 | 7 | 217 | 0 | 7 | 217 | 56 | 11 | 231 | 525 |
| 66 | 4 | 57 | 0 | 4 | 57 | 78 | 5 | 57 | 23 |
| 67 | 3 | 102 | 2 | 3 | 102 | 66 | 5 | 142 | 104 |
| 68 | 8 | 103 | 0 | 5 | 103 | 71 | 11 | 137 | 2649 |
| 69 | 11 | 201 | 1 | 7 | 201 | 57 | 13 | 205 | 38623 |
| 70 | 7 | 117 | 1 | 7 | 117 | 58 | 10 | 127 | 8275 |
| 71 | 7 | 258 | 0 | 7 | 258 | 55 | 8 | 264 | 140168 |
| 72 | 3 | 49 | 0 | 3 | 49 | 59 | 4 | 65 | 18 |
| 73 | 7 | 223 | 1 | 6 | 223 | 61 | 13 | 231 | 29945 |
| 74 | 4 | 109 | 3 | 4 | 109 | 74 | 11 | 165 | 2598 |
| 75 | 7 | 162 | 8 | 6 | 162 | 74 | 9 | 188 | 5690 |
| 76 | 4 | 33 | 0 | 4 | 33 | 54 | 5 | 33 | 11 |
| 77 | 6 | 196 | 0 | 6 | 196 | 62 | 9 | 228 | 109487 |
| 78 | 7 | 112 | 0 | 7 | 112 | 59 | 7 | 114 | 147 |
| 79 | 4 | 66 | 0 | 3 | 66 | 52 | 8 | 130 | 236 |
| 80 | 1 | 11 | 0 | 1 | 11 | 56 | 2 | 13 | 2 |
| Suma | 509 | 10436 | 30 | 453 | 10436 | 4865 | 665 | 12440 | 1766570 |
| Prom | 6.3625 | 130.45 | 2996.375 | 5.6625 | 130.45 | 3056.813 | 8.3125 | 155.5 | 22082.13 |
| ValMin | 1 | 11 | 2996 | 1 | 11 | 3042 | 2 | 11 | 1 |
| ValMax | 18 | 258 | 3004 | 16 | 258 | 3111 | 13 | 318 | 148510 |
| DesvEst | 3.424571 | 70.76399 | 1.023459 | 2.920654 | 70.76399 | 11.33282 | 2.754025 | 73.31741 | 42387.6 |

Tabla A.9: Ambiente 5. Resolución: 20cm. Propagación de frente de onda: 2999ms

| Prueba | SVP | | | SVA | | | MA | | |
|---------|-------------------|----------|----------|-------------------|----------|----------|-------------------|----------|--------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 17 | 135 | 0 | 60 | 135 | 0 | 4 | 135 | 0 |
| 2 | 5 | 93 | 0 | 19 | 93 | 0 | 5 | 93 | 0 |
| 3 | 9 | 146 | 0 | 78 | 146 | 0 | 6 | 146 | 0 |
| 4 | 29 | 167 | 0 | 89 | 167 | 0 | 5 | 167 | 0 |
| 5 | 3 | 129 | 1 | 50 | 129 | 0 | 3 | 129 | 0 |
| 6 | 3 | 86 | 0 | 63 | 86 | 0 | 3 | 86 | 0 |
| 7 | 5 | 110 | 0 | 22 | 110 | 0 | 5 | 110 | 0 |
| 8 | 4 | 39 | 0 | 21 | 39 | 0 | 5 | 39 | 0 |
| 9 | 24 | 137 | 0 | 37 | 137 | 0 | 7 | 137 | 0 |
| 10 | 3 | 80 | 0 | 47 | 80 | 0 | 3 | 80 | 0 |
| 11 | 18 | 137 | 0 | 38 | 137 | 0 | 5 | 137 | 0 |
| 12 | 23 | 91 | 0 | 55 | 91 | 0 | 9 | 91 | 0 |
| 13 | 2 | 37 | 0 | 10 | 37 | 0 | 2 | 37 | 0 |
| 14 | 9 | 108 | 0 | 44 | 108 | 0 | 4 | 108 | 0 |
| 15 | 8 | 97 | 0 | 40 | 97 | 0 | 5 | 97 | 0 |
| 16 | 24 | 137 | 0 | 32 | 137 | 4 | 7 | 137 | 0 |
| 17 | 3 | 74 | 0 | 39 | 74 | 0 | 3 | 74 | 0 |
| 18 | 4 | 82 | 0 | 37 | 82 | 0 | 4 | 82 | 0 |
| 19 | 3 | 62 | 0 | 26 | 62 | 0 | 4 | 62 | 0 |
| 20 | 4 | 55 | 0 | 21 | 55 | 0 | 5 | 55 | 0 |
| 21 | 3 | 56 | 0 | 30 | 56 | 0 | 3 | 56 | 0 |
| 22 | 3 | 71 | 0 | 39 | 71 | 0 | 4 | 71 | 0 |
| 23 | 27 | 151 | 0 | 70 | 151 | 0 | 5 | 151 | 0 |
| 24 | 3 | 109 | 0 | 39 | 109 | 0 | 3 | 109 | 0 |
| 25 | 26 | 122 | 0 | 37 | 122 | 0 | 5 | 122 | 0 |
| 26 | 2 | 47 | 0 | 16 | 47 | 0 | 2 | 47 | 0 |
| 27 | 27 | 156 | 0 | 50 | 156 | 0 | 7 | 156 | 0 |
| 28 | 30 | 169 | 0 | 70 | 169 | 0 | 7 | 169 | 0 |
| 29 | 37 | 148 | 0 | 70 | 148 | 0 | 8 | 148 | 0 |
| 30 | 17 | 111 | 0 | 38 | 111 | 0 | 5 | 111 | 0 |
| 31 | 4 | 51 | 0 | 13 | 51 | 0 | 4 | 51 | 0 |
| 32 | 19 | 118 | 0 | 49 | 118 | 0 | 11 | 118 | 0 |
| 33 | 4 | 42 | 0 | 22 | 42 | 0 | 4 | 42 | 0 |
| 34 | 28 | 110 | 0 | 74 | 110 | 0 | 5 | 110 | 0 |
| 35 | 4 | 61 | 0 | 45 | 61 | 0 | 4 | 61 | 0 |
| 36 | 3 | 95 | 0 | 22 | 95 | 0 | 3 | 95 | 0 |
| 37 | 3 | 56 | 0 | 27 | 56 | 0 | 3 | 56 | 0 |
| 38 | 3 | 122 | 0 | 42 | 122 | 0 | 3 | 122 | 0 |
| 39 | 3 | 102 | 0 | 63 | 102 | 0 | 3 | 102 | 0 |
| 40 | 19 | 159 | 0 | 94 | 159 | 0 | 5 | 159 | 0 |
| 41 | 4 | 22 | 0 | 14 | 22 | 0 | 4 | 22 | 0 |
| 42 | 3 | 69 | 0 | 30 | 69 | 0 | 3 | 69 | 0 |
| 43 | 3 | 76 | 0 | 20 | 76 | 0 | 3 | 76 | 0 |
| 44 | 8 | 88 | 0 | 51 | 88 | 0 | 5 | 88 | 0 |
| 45 | 5 | 106 | 0 | 42 | 106 | 0 | 4 | 106 | 0 |
| 46 | 11 | 118 | 0 | 45 | 118 | 0 | 4 | 118 | 0 |
| 47 | 16 | 67 | 0 | 17 | 67 | 0 | 3 | 67 | 0 |
| 48 | 3 | 90 | 0 | 51 | 90 | 0 | 3 | 90 | 0 |
| 49 | 17 | 154 | 0 | 66 | 154 | 0 | 6 | 154 | 0 |
| 50 | 11 | 130 | 0 | 65 | 130 | 0 | 7 | 130 | 0 |
| 51 | 19 | 140 | 0 | 42 | 140 | 6 | 5 | 140 | 0 |
| 52 | 37 | 129 | 0 | 67 | 129 | 0 | 10 | 129 | 0 |
| 53 | 17 | 121 | 0 | 46 | 121 | 0 | 5 | 121 | 0 |
| 54 | 4 | 62 | 0 | 39 | 62 | 0 | 4 | 62 | 0 |
| 55 | 5 | 100 | 0 | 47 | 100 | 0 | 4 | 100 | 0 |
| 56 | 30 | 131 | 0 | 66 | 131 | 0 | 7 | 131 | 0 |
| 57 | 4 | 28 | 0 | 13 | 28 | 0 | 4 | 28 | 0 |
| 58 | 29 | 127 | 0 | 54 | 127 | 0 | 4 | 127 | 0 |
| 59 | 22 | 89 | 0 | 31 | 89 | 0 | 4 | 89 | 0 |
| 60 | 3 | 27 | 0 | 14 | 27 | 0 | 3 | 27 | 0 |
| 61 | 10 | 94 | 0 | 36 | 94 | 0 | 6 | 94 | 0 |
| 62 | 37 | 161 | 0 | 84 | 161 | 0 | 9 | 161 | 0 |
| 63 | 4 | 53 | 0 | 19 | 53 | 0 | 4 | 53 | 0 |
| 64 | 18 | 122 | 0 | 44 | 122 | 0 | 4 | 122 | 0 |
| 65 | 3 | 101 | 0 | 47 | 101 | 0 | 3 | 101 | 0 |
| 66 | 8 | 111 | 0 | 48 | 111 | 0 | 5 | 111 | 0 |
| 67 | 5 | 120 | 0 | 33 | 120 | 0 | 4 | 120 | 0 |
| 68 | 3 | 37 | 0 | 6 | 37 | 0 | 3 | 37 | 0 |
| 69 | 3 | 79 | 0 | 41 | 79 | 0 | 3 | 79 | 0 |
| 70 | 20 | 142 | 0 | 64 | 142 | 0 | 7 | 142 | 0 |
| 71 | 2 | 29 | 0 | 13 | 29 | 0 | 2 | 29 | 0 |
| 72 | 15 | 155 | 0 | 67 | 155 | 0 | 5 | 155 | 0 |
| 73 | 37 | 133 | 0 | 61 | 133 | 0 | 8 | 133 | 0 |
| 74 | 4 | 20 | 0 | 12 | 20 | 0 | 4 | 20 | 0 |
| 75 | 36 | 163 | 0 | 78 | 163 | 0 | 7 | 163 | 0 |
| 76 | 10 | 114 | 0 | 53 | 114 | 0 | 5 | 114 | 0 |
| 77 | 19 | 133 | 0 | 77 | 133 | 0 | 4 | 133 | 0 |
| 78 | 4 | 74 | 2 | 52 | 74 | 0 | 4 | 74 | 0 |
| 79 | 41 | 132 | 0 | 55 | 132 | 0 | 6 | 132 | 0 |
| 80 | 2 | 11 | 0 | 7 | 11 | 0 | 2 | 11 | 0 |
| Suma | 997 | 7916 | 3 | 3455 | 7916 | 10 | 375 | 7916 | 0 |
| Prom | 12.4625 | 98.95 | 29999.04 | 43.1875 | 98.95 | 29999.13 | 4.6875 | 98.95 | 29999 |
| ValMin | 2 | 11 | 0 | 6 | 11 | 0 | 2 | 11 | 0 |
| ValMax | 41 | 169 | 2 | 94 | 169 | 6 | 11 | 169 | 0 |
| DesvEst | 11.26318 | 40.84825 | 0.248731 | 20.94302 | 40.84825 | 0.801502 | 1.845924 | 40.84825 | 0 |

Tabla A.10: Ambiente 5. Resolución: 20cm. Propagación de frente de onda: 2999ms

| Prueba | MAVC | | | GR | | | A*RR | | |
|---------|-------------------|----------|--------|-------------------|----------|----------|-------------------|----------|----------|
| | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo | Reconfiguraciones | Longitud | Tiempo |
| 1 | 5 | 135 | 0 | 4 | 135 | 62 | 6 | 149 | 110 |
| 2 | 6 | 93 | 0 | 4 | 93 | 62 | 7 | 153 | 109 |
| 3 | 7 | 146 | 0 | 6 | 146 | 47 | 7 | 146 | 1045 |
| 4 | 6 | 167 | 0 | 5 | 167 | 63 | 8 | 201 | 328 |
| 5 | 3 | 129 | 0 | 3 | 129 | 46 | 5 | 135 | 78 |
| 6 | 4 | 86 | 0 | 3 | 86 | 46 | 5 | 108 | 46 |
| 7 | 5 | 110 | 0 | 4 | 110 | 63 | 10 | 124 | 1014 |
| 8 | 5 | 39 | 0 | 4 | 39 | 47 | 6 | 43 | 125 |
| 9 | 7 | 137 | 0 | 6 | 137 | 62 | 9 | 145 | 4368 |
| 10 | 3 | 80 | 0 | 3 | 80 | 47 | 5 | 94 | 47 |
| 11 | 5 | 137 | 0 | 5 | 137 | 46 | 10 | 173 | 132584 |
| 12 | 8 | 91 | 0 | 6 | 91 | 78 | 9 | 91 | 655 |
| 13 | 2 | 37 | 0 | 2 | 37 | 46 | 4 | 67 | 15 |
| 14 | 6 | 108 | 0 | 4 | 108 | 63 | 6 | 108 | 546 |
| 15 | 6 | 97 | 0 | 5 | 97 | 63 | 7 | 97 | 8174 |
| 16 | 7 | 137 | 0 | 6 | 137 | 47 | 9 | 145 | 4290 |
| 17 | 3 | 74 | 0 | 3 | 74 | 47 | 7 | 90 | 78 |
| 18 | 4 | 82 | 0 | 4 | 82 | 63 | 5 | 82 | 31 |
| 19 | 3 | 62 | 0 | 3 | 62 | 63 | 6 | 102 | 63 |
| 20 | 5 | 55 | 0 | 4 | 55 | 62 | 5 | 55 | 62 |
| 21 | 3 | 56 | 0 | 3 | 56 | 47 | 5 | 118 | 62 |
| 22 | 4 | 71 | 0 | 3 | 71 | 62 | 6 | 93 | 78 |
| 23 | 6 | 151 | 0 | 4 | 151 | 47 | 8 | 183 | 827 |
| 24 | 4 | 109 | 0 | 3 | 109 | 47 | 5 | 115 | 62 |
| 25 | 5 | 122 | 0 | 5 | 122 | 63 | 8 | 140 | 1654 |
| 26 | 2 | 47 | 0 | 2 | 47 | 47 | 3 | 47 | 16 |
| 27 | 5 | 156 | 0 | 5 | 156 | 62 | 10 | 164 | 6271 |
| 28 | 8 | 169 | 0 | 6 | 169 | 47 | 8 | 189 | 359 |
| 29 | 10 | 148 | 0 | 8 | 148 | 63 | 8 | 166 | 203 |
| 30 | 4 | 111 | 0 | 4 | 111 | 62 | 7 | 153 | 187 |
| 31 | 4 | 51 | 0 | 4 | 51 | 62 | 5 | 51 | 78 |
| 32 | 10 | 118 | 0 | 8 | 118 | 47 | 9 | 170 | 8767 |
| 33 | 4 | 42 | 0 | 4 | 42 | 63 | 5 | 42 | 16 |
| 34 | 5 | 110 | 0 | 5 | 110 | 47 | 6 | 110 | 62 |
| 35 | 4 | 61 | 0 | 4 | 61 | 47 | 6 | 61 | 31 |
| 36 | 3 | 95 | 0 | 3 | 95 | 46 | 5 | 139 | 78 |
| 37 | 3 | 56 | 0 | 3 | 56 | 62 | 6 | 88 | 47 |
| 38 | 3 | 122 | 0 | 3 | 122 | 62 | 5 | 132 | 78 |
| 39 | 3 | 102 | 0 | 3 | 102 | 47 | 5 | 122 | 62 |
| 40 | 5 | 159 | 0 | 5 | 159 | 63 | 6 | 159 | 453 |
| 41 | 4 | 22 | 0 | 4 | 22 | 62 | 4 | 22 | 62 |
| 42 | 3 | 69 | 0 | 3 | 69 | 47 | 5 | 125 | 63 |
| 43 | 3 | 76 | 0 | 3 | 76 | 47 | 5 | 118 | 62 |
| 44 | 5 | 88 | 0 | 5 | 88 | 47 | 5 | 88 | 47 |
| 45 | 4 | 106 | 0 | 4 | 106 | 47 | 7 | 110 | 78 |
| 46 | 4 | 118 | 0 | 4 | 118 | 62 | 7 | 186 | 202 |
| 47 | 3 | 67 | 0 | 3 | 67 | 47 | 6 | 69 | 31 |
| 48 | 3 | 90 | 0 | 3 | 90 | 62 | 5 | 104 | 47 |
| 49 | 6 | 154 | 0 | 6 | 154 | 62 | 7 | 154 | 1092 |
| 50 | 7 | 130 | 0 | 7 | 130 | 62 | 6 | 134 | 296 |
| 51 | 5 | 140 | 0 | 5 | 140 | 47 | 6 | 140 | 94 |
| 52 | 11 | 129 | 0 | 9 | 129 | 62 | 12 | 215 | 147295 |
| 53 | 5 | 121 | 0 | 4 | 121 | 47 | 8 | 163 | 796 |
| 54 | 4 | 62 | 0 | 4 | 62 | 47 | 5 | 62 | 31 |
| 55 | 4 | 100 | 0 | 4 | 100 | 47 | 6 | 164 | 110 |
| 56 | 8 | 131 | 0 | 6 | 131 | 62 | 9 | 179 | 3588 |
| 57 | 4 | 28 | 0 | 4 | 28 | 46 | 5 | 28 | 46 |
| 58 | 4 | 127 | 0 | 4 | 127 | 47 | 7 | 157 | 125 |
| 59 | 4 | 89 | 0 | 4 | 89 | 62 | 6 | 89 | 47 |
| 60 | 3 | 27 | 0 | 3 | 27 | 62 | 4 | 27 | 16 |
| 61 | 5 | 94 | 0 | 5 | 94 | 47 | 6 | 94 | 47 |
| 62 | 9 | 161 | 0 | 7 | 161 | 62 | 7 | 207 | 920 |
| 63 | 4 | 53 | 0 | 4 | 53 | 47 | 5 | 53 | 16 |
| 64 | 5 | 122 | 0 | 4 | 122 | 47 | 9 | 144 | 890 |
| 65 | 4 | 101 | 0 | 3 | 101 | 46 | 5 | 113 | 62 |
| 66 | 5 | 111 | 0 | 5 | 111 | 47 | 7 | 111 | 78 |
| 67 | 4 | 120 | 0 | 4 | 120 | 47 | 10 | 214 | 125096 |
| 68 | 3 | 37 | 0 | 3 | 37 | 63 | 4 | 37 | 16 |
| 69 | 3 | 79 | 0 | 3 | 79 | 47 | 5 | 95 | 47 |
| 70 | 7 | 142 | 0 | 7 | 142 | 47 | 10 | 168 | 52837 |
| 71 | 2 | 29 | 0 | 2 | 29 | 62 | 3 | 65 | 15 |
| 72 | 5 | 155 | 0 | 5 | 155 | 63 | 9 | 211 | 1748 |
| 73 | 8 | 133 | 0 | 6 | 133 | 62 | 8 | 193 | 3400 |
| 74 | 4 | 20 | 0 | 4 | 20 | 63 | 4 | 20 | 78 |
| 75 | 7 | 163 | 0 | 6 | 163 | 47 | 9 | 163 | 110 |
| 76 | 5 | 114 | 0 | 5 | 114 | 62 | 6 | 114 | 234 |
| 77 | 5 | 133 | 0 | 4 | 133 | 46 | 6 | 141 | 78 |
| 78 | 4 | 74 | 0 | 4 | 74 | 46 | 5 | 74 | 31 |
| 79 | 5 | 132 | 0 | 5 | 132 | 63 | 10 | 192 | 106888 |
| 80 | 2 | 11 | 0 | 2 | 11 | 47 | 4 | 73 | 46 |
| Suma | 385 | 7916 | 0 | 346 | 7916 | 4351 | 519 | 9566 | 619924 |
| Prom | 4.8125 | 98.95 | 29999 | 4.325 | 98.95 | 30053.39 | 6.4875 | 119.575 | 7749.05 |
| ValMin | 2 | 11 | 0 | 2 | 11 | 46 | 3 | 20 | 15 |
| ValMax | 11 | 169 | 0 | 9 | 169 | 78 | 12 | 215 | 147295 |
| DesvEst | 1.90332 | 40.84825 | 0 | 1.430013 | 40.84825 | 8.239291 | 1.909296 | 50.4586 | 28594.71 |

Tabla A.11: Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms

| Prueba | SVA | | | SVP | | |
|----------------|-----------------|-----------------|------------|-----------------|-----------------|-----------------|
| | Reconf. | Longitud | Tiempo | Reconf. | Longitud | Tiempo |
| 1 | 24 | 71 | 0 | 34 | 71 | 0 |
| 2 | 24 | 63 | 0 | 38 | 63 | 0 |
| 3 | 4 | 39 | 0 | 16 | 39 | 0 |
| 4 | 18 | 43 | 0 | 16 | 43 | 0 |
| 5 | 10 | 45 | 0 | 22 | 45 | 0 |
| 6 | 13 | 75 | 0 | 14 | 75 | 0 |
| 7 | 25 | 104 | 0 | 32 | 104 | 0 |
| 8 | 25 | 64 | 0 | 30 | 64 | 1 |
| 9 | 11 | 70 | 0 | 19 | 70 | 0 |
| 10 | 4 | 42 | 0 | 12 | 42 | 0 |
| 11 | 5 | 34 | 0 | 11 | 34 | 0 |
| 12 | 23 | 93 | 0 | 31 | 93 | 0 |
| 13 | 3 | 18 | 0 | 8 | 18 | 0 |
| 14 | 4 | 21 | 0 | 14 | 21 | 0 |
| 15 | 9 | 38 | 0 | 9 | 38 | 0 |
| 16 | 21 | 98 | 0 | 45 | 98 | 0 |
| 17 | 7 | 52 | 0 | 25 | 52 | 0 |
| 18 | 17 | 46 | 0 | 23 | 46 | 0 |
| 19 | 12 | 98 | 0 | 18 | 98 | 0 |
| 20 | 9 | 78 | 0 | 29 | 78 | 0 |
| 21 | 13 | 98 | 0 | 32 | 98 | 0 |
| 22 | 13 | 51 | 0 | 20 | 51 | 0 |
| 23 | 4 | 9 | 0 | 5 | 9 | 0 |
| 24 | 9 | 78 | 0 | 29 | 78 | 0 |
| 25 | 19 | 59 | 0 | 30 | 59 | 0 |
| 26 | 17 | 62 | 0 | 25 | 62 | 2 |
| 27 | 27 | 81 | 0 | 25 | 81 | 0 |
| 28 | 3 | 47 | 0 | 16 | 47 | 1 |
| 29 | 21 | 71 | 0 | 20 | 71 | 0 |
| 30 | 4 | 36 | 0 | 10 | 36 | 0 |
| 31 | 9 | 30 | 0 | 15 | 30 | 0 |
| 32 | 21 | 77 | 0 | 26 | 77 | 0 |
| 33 | 10 | 74 | 0 | 40 | 74 | 0 |
| 34 | 10 | 71 | 0 | 20 | 71 | 0 |
| 35 | 4 | 30 | 0 | 20 | 30 | 0 |
| 36 | 13 | 75 | 0 | 31 | 75 | 0 |
| 37 | 19 | 66 | 0 | 27 | 66 | 0 |
| 38 | 3 | 39 | 0 | 13 | 39 | 0 |
| 39 | 3 | 30 | 0 | 6 | 30 | 0 |
| 40 | 23 | 129 | 0 | 51 | 129 | 0 |
| 41 | 3 | 30 | 0 | 17 | 30 | 0 |
| 42 | 23 | 69 | 0 | 29 | 69 | 0 |
| 43 | 4 | 34 | 0 | 14 | 34 | 0 |
| 44 | 10 | 34 | 0 | 7 | 34 | 0 |
| 45 | 27 | 101 | 0 | 33 | 101 | 0 |
| 46 | 2 | 10 | 0 | 4 | 10 | 0 |
| 47 | 3 | 38 | 0 | 25 | 38 | 0 |
| 48 | 3 | 23 | 0 | 15 | 23 | 0 |
| 49 | 4 | 14 | 0 | 9 | 14 | 0 |
| 50 | 16 | 111 | 0 | 44 | 111 | 0 |
| Total | 608 | 2869 | 0 | 1104 | 2869 | 4 |
| Prom | 12.16 | 57.38 | 375 | 22.08 | 57.38 | 375.08 |
| ValMin | 2 | 9 | 0 | 4 | 9 | 0 |
| ValMax | 27 | 129 | 0 | 51 | 129 | 2 |
| DesvEst | 8.069494 | 28.55849 | 0 | 11.02102 | 28.55849 | 0.340468 |

Tabla A.12: Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms

| Prueba | MA | | | MAVC | | | GR | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Reconf. | Longitud | Tiempo | Reconf. | Longitud | Tiempo | Reconf. | Longitud | Tiempo |
| 1 | 7 | 71 | 0 | 7 | 71 | 1 | 7 | 71 | 47 |
| 2 | 7 | 63 | 0 | 7 | 63 | 0 | 6 | 63 | 47 |
| 3 | 5 | 39 | 0 | 5 | 39 | 0 | 4 | 39 | 31 |
| 4 | 5 | 43 | 0 | 5 | 43 | 0 | 5 | 43 | 47 |
| 5 | 7 | 45 | 0 | 7 | 45 | 0 | 6 | 45 | 62 |
| 6 | 8 | 75 | 0 | 7 | 75 | 0 | 7 | 75 | 63 |
| 7 | 10 | 104 | 0 | 10 | 104 | 0 | 9 | 104 | 63 |
| 8 | 6 | 64 | 0 | 6 | 64 | 0 | 6 | 64 | 47 |
| 9 | 7 | 70 | 0 | 7 | 70 | 0 | 6 | 70 | 47 |
| 10 | 4 | 42 | 0 | 4 | 42 | 0 | 4 | 42 | 47 |
| 11 | 3 | 34 | 0 | 3 | 34 | 0 | 3 | 34 | 47 |
| 12 | 12 | 93 | 0 | 13 | 93 | 0 | 10 | 93 | 47 |
| 13 | 3 | 18 | 0 | 3 | 18 | 0 | 3 | 18 | 63 |
| 14 | 4 | 21 | 0 | 4 | 21 | 0 | 4 | 21 | 47 |
| 15 | 6 | 38 | 0 | 6 | 38 | 0 | 6 | 38 | 47 |
| 16 | 6 | 98 | 0 | 6 | 98 | 0 | 5 | 98 | 47 |
| 17 | 4 | 52 | 0 | 4 | 52 | 0 | 3 | 52 | 63 |
| 18 | 5 | 46 | 0 | 5 | 46 | 0 | 5 | 46 | 47 |
| 19 | 6 | 98 | 0 | 6 | 98 | 0 | 6 | 98 | 31 |
| 20 | 7 | 78 | 0 | 7 | 78 | 0 | 5 | 78 | 47 |
| 21 | 7 | 98 | 0 | 7 | 98 | 0 | 7 | 98 | 31 |
| 22 | 6 | 51 | 0 | 6 | 51 | 0 | 5 | 51 | 47 |
| 23 | 4 | 9 | 0 | 4 | 9 | 0 | 4 | 9 | 46 |
| 24 | 6 | 78 | 0 | 6 | 78 | 0 | 5 | 78 | 46 |
| 25 | 9 | 59 | 0 | 9 | 59 | 0 | 8 | 59 | 47 |
| 26 | 7 | 62 | 0 | 7 | 62 | 0 | 6 | 62 | 31 |
| 27 | 10 | 81 | 0 | 10 | 81 | 0 | 4 | 81 | 47 |
| 28 | 3 | 47 | 0 | 3 | 47 | 0 | 3 | 47 | 47 |
| 29 | 5 | 71 | 0 | 5 | 71 | 0 | 5 | 71 | 47 |
| 30 | 6 | 36 | 0 | 5 | 36 | 0 | 4 | 36 | 47 |
| 31 | 3 | 30 | 0 | 4 | 30 | 0 | 3 | 30 | 31 |
| 32 | 8 | 77 | 0 | 8 | 77 | 0 | 7 | 77 | 63 |
| 33 | 11 | 74 | 0 | 10 | 74 | 0 | 9 | 74 | 47 |
| 34 | 6 | 71 | 0 | 6 | 71 | 0 | 6 | 71 | 31 |
| 35 | 4 | 30 | 0 | 4 | 30 | 0 | 4 | 30 | 47 |
| 36 | 7 | 75 | 0 | 9 | 75 | 0 | 7 | 75 | 47 |
| 37 | 5 | 66 | 0 | 5 | 66 | 0 | 5 | 66 | 62 |
| 38 | 4 | 39 | 0 | 4 | 39 | 0 | 3 | 39 | 47 |
| 39 | 3 | 30 | 0 | 3 | 30 | 0 | 3 | 30 | 47 |
| 40 | 7 | 129 | 0 | 7 | 129 | 0 | 6 | 129 | 32 |
| 41 | 3 | 30 | 0 | 3 | 30 | 0 | 3 | 30 | 47 |
| 42 | 10 | 69 | 0 | 10 | 69 | 0 | 8 | 69 | 47 |
| 43 | 4 | 34 | 0 | 5 | 34 | 0 | 4 | 34 | 47 |
| 44 | 3 | 34 | 0 | 3 | 34 | 0 | 3 | 34 | 31 |
| 45 | 8 | 101 | 0 | 8 | 101 | 0 | 8 | 101 | 47 |
| 46 | 2 | 10 | 0 | 2 | 10 | 0 | 2 | 10 | 32 |
| 47 | 3 | 38 | 0 | 5 | 38 | 0 | 3 | 38 | 31 |
| 48 | 3 | 23 | 1 | 3 | 23 | 0 | 3 | 23 | 31 |
| 49 | 4 | 14 | 0 | 4 | 14 | 0 | 4 | 14 | 63 |
| 50 | 9 | 111 | 0 | 10 | 111 | 0 | 8 | 111 | 47 |
| Total | 292 | 2869 | 1 | 297 | 2869 | 1 | 260 | 2869 | 2300 |
| Prom | 5.84 | 57.38 | 375.02 | 5.94 | 57.38 | 375.02 | 5.2 | 57.38 | 421 |
| ValMin | 2 | 9 | 0 | 2 | 9 | 0 | 2 | 9 | 31 |
| ValMax | 12 | 129 | 0 | 13 | 129 | 1 | 10 | 129 | 63 |
| DesvEst | 2.393529 | 28.55849 | 0.141421 | 2.402465 | 28.55849 | 0.141421 | 1.927248 | 28.55849 | 9.785454 |

Tabla A.13: Ambiente 4. Resolución: 50cm. Propagación de frente de onda: 375ms

| Prueba | A* | | | A*RR | | |
|----------------|-----------------|-----------------|-----------------|-----------------|----------------|-----------------|
| | Reconf. | Longitud | Tiempo | Reconf. | Longitud | Tiempo |
| 1 | 16 | 71 | 1531 | 9 | 73 | 1375 |
| 2 | 16 | 63 | 844 | 6 | 65 | 109 |
| 3 | 5 | 39 | 390 | 6 | 41 | 156 |
| 4 | 8 | 43 | 78 | 5 | 45 | 63 |
| 5 | 11 | 45 | 359 | 8 | 45 | 328 |
| 6 | 12 | 75 | 1422 | 8 | 95 | 7875 |
| 7 | 40 | 105 | 36422 | 10 | 142 | 5516 |
| 8 | 17 | 64 | 781 | 8 | 66 | 250 |
| 9 | 17 | 70 | 5047 | 8 | 70 | 3515 |
| 10 | 5 | 42 | 453 | 5 | 42 | 47 |
| 11 | 5 | 34 | 32 | 4 | 34 | 15 |
| 12 | 21 | 93 | 272625 | 14 | 107 | 5266 |
| 13 | 9 | 18 | 2282 | 6 | 20 | 171 |
| 14 | 5 | 21 | 125 | 4 | 21 | 31 |
| 15 | 8 | 38 | 93 | 7 | 42 | 313 |
| 16 | 15 | 99 | 87890 | 8 | 108 | 109 |
| 17 | 7 | 53 | 8531 | 5 | 58 | 31 |
| 18 | 10 | 46 | 500 | 6 | 46 | 1203 |
| 19 | 10 | 98 | 5110 | 7 | 98 | 2171 |
| 20 | 15 | 79 | 198906 | 7 | 122 | 94 |
| 21 | 33 | 100 | 16094 | 9 | 118 | 766 |
| 22 | 22 | 51 | 5062 | 5 | 91 | 110 |
| 23 | 5 | 9 | 40 | 4 | 9 | 110 |
| 24 | 28 | 79 | 70594 | 9 | 126 | 735 |
| 25 | 18 | 59 | 1141 | 7 | 65 | 93 |
| 26 | 19 | 62 | 1703 | 8 | 102 | 609 |
| 27 | 11 | 82 | 129391 | 8 | 143 | 2094 |
| 28 | 4 | 47 | 641 | 4 | 47 | 16 |
| 29 | 18 | 71 | 25468 | 8 | 123 | 4578 |
| 30 | 5 | 36 | 594 | 5 | 36 | 78 |
| 31 | 13 | 31 | 49719 | 6 | 96 | 78 |
| 32 | 22 | 77 | 7203 | 9 | 81 | 2437 |
| 33 | 11 | 74 | 6453 | 9 | 76 | 1859 |
| 34 | 11 | 71 | 5360 | 7 | 71 | 63 |
| 35 | 5 | 30 | 234 | 5 | 30 | 31 |
| 36 | 14 | 75 | 12562 | 7 | 75 | 78 |
| 37 | 21 | 67 | 8406 | 9 | 100 | 422 |
| 38 | 7 | 40 | 1343 | 6 | 71 | 391 |
| 39 | 4 | 30 | 31 | 3 | 30 | 15 |
| 40 | 18 | 129 | 98750 | 13 | 135 | 328 |
| 41 | 9 | 31 | 3735 | 8 | 86 | 485 |
| 42 | 10 | 69 | 3953 | 8 | 69 | 1781 |
| 43 | 5 | 34 | 453 | 5 | 34 | 15 |
| 44 | 14 | 34 | 21032 | 8 | 102 | 188 |
| 45 | 40 | 102 | 16609 | 10 | 127 | 2094 |
| 46 | 3 | 10 | 1 | 3 | 10 | 1 |
| 47 | 7 | 39 | 27422 | 7 | 80 | 438 |
| 48 | 6 | 23 | 1719 | 5 | 71 | 62 |
| 49 | 5 | 14 | 31 | 4 | 14 | 62 |
| 50 | 15 | 111 | 108250 | 9 | 115 | 10062 |
| Total | 655 | 2883 | 1247415 | 349 | 3673 | 58717 |
| Prom | 13.1 | 57.66 | 24948.3 | 6.98 | 73.46 | 1174.34 |
| ValMin | 3 | 9 | 1 | 3 | 9 | 1 |
| ValMax | 40 | 129 | 272625 | 14 | 143 | 10062 |
| DesvEst | 8.652592 | 28.69659 | 53531.45 | 2.290085 | 36.8351 | 2109.209 |

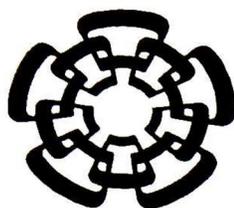
Bibliografía

- [1] Rodney A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *MIT Artificial Intelligence Laboratory. A.I. Memo 1091*, 1989.
- [2] Rodney A. Brooks. *Cuerpos y Máquinas. De los robots humanos a los hombres robots*. Ediciones B, 2003.
- [3] Luis Ignacio Díaz del Dedo and Luis Alberto García Pérez. Estado del arte en robótica cuadrúpeda semiautónoma, 2007. Disponible en <http://www.r4p.es/PFC/docs/Estado%20de%20el%20Arte%20Rob%F3tica%20Cuadrupeda.pdf>. Última vez consultado el 4 de Noviembre de 2008.
- [4] Lynne E. Parker and Fang Tang. Building multirobot coalitions through automated task solution synthesis. *Proceedings of the IEEE*, 94(7):1289–1305, 2006.
- [5] Bill Gates. A robot in every home. <http://www.sciam.com/article.cfm?id=a-robot-in-every-home>. Última vez consultada el 4 de Noviembre de 2008.
- [6] Eu Jan Gene, Juang Tong-Ying, Huang Jun-Da, Su Chien-Min, and Cheng Chih-Yung. A fast path planning algorithm for piano mover's problem on raster. *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pages 522–527, 2005.
- [7] Brian P. Gerkey and Maja J Matari. Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 1:464–469, 2002.

- [8] P.E. Hart, N.J. Nilsson, and B Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 2:100–107, 1968.
- [9] Honda. Asimo. <http://asimo.honda.com>. Última vez consultada el 4 de Noviembre de 2008.
- [10] Lengyel Jed, Reichert Mark, R. Donald Bruce, and P. Greenberg Donald. Real-time robot motion planning using rasterizing computer graphics hardware. *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 327–335, 1990.
- [11] Kresimir Kovac, Ivan Zivkovic, and Borjana Dalbelo Basic. Simulation of multi-robot reinforcement learning for box-pushing problem. *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, 2:603–606, 2004.
- [12] Steven M. LaValle. *Planning Algorithms*. Cambridge, 2006. Disponible en <http://planning.cs.uiuc.edu/>. Última vez consultado el 4 de Noviembre de 2008.
- [13] Ho-Gil Lee. [eleconomista.es](http://www.eleconomista.es/), 2006. <http://www.eleconomista.es/index.html>. Última vez consultada el 4 de Noviembre de 2008.
- [14] Yangmin Li and Xin Chen. Modeling and simulation of a swarm of robots for box-pushing task. *12th Mediterranean Conference on Control and Automation, Kusadasi, Aydin, Turkey, 2004*.
- [15] Tomer Livneh. The “piano movers problem”, 2006. Disponible en <http://www.math.tau.ac.il/~mvplab/Completed%20Projects/Piano%20Movers/Piano%20movers.pdf>. Última vez consultado el 4 de Noviembre de 2008.

-
- [16] Tomás Lozano-Pérez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, pages 108–120, 1983.
- [17] Angelica Muñoz Melendez and Alexis Drogoul. Analyzing multi-robot box-pushing. *Avances en la Ciencia de la Computación. Memoria de los Talleres del Quinto Encuentro Internacional de Computación ENC'04. Universidad de Colima*, 1:530–539, 2004.
- [18] Gabriel Ramírez and Said Zegloul. A new local path planner for nonholonomic mobile robot navigation in cluttered environments. *ICRA 2000: Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, 3:2058–2063, 2000.
- [19] J. H. Reif. Complexity of the mover's problem and generalizations. *Annual IEEE Symposium on Foundations of Computer Science*, pages 421–427, 1979.
- [20] J.H. Reif and M. Sharir. Motion planning in the presence of moving obstacles. *26th Annual IEEE Symposium on Foundations of Computer Science*, pages 144–154, 1985.
- [21] Kamran H. SEDIGHI, Kaveh ASHENAYI, Theodore W. MANIKAS, Roger L. WAINWRIGHT, and Heng-Ming TAI. Autonomous local path planning for a mobile robot using a genetic algorithm. *CEC 2004: Proceedings of the 2004 congress on evolutionary computation*, pages 1338–1345, 2004.
- [22] Sony. Aibo. <http://support.sony-europe.com/aibo/>. Última vez consultada el 4 de Noviembre de 2008.
- [23] Sony. Qrio. <http://www.sonyaibo.net/aboutqrio.htm>. Última vez consultada el 4 de Noviembre de 2008.

- [24] Simeon T, Laumond J-P, and Nissoux C. Visibility based probabilistic roadmaps for motion planning. *Submitted to the Advanced Robotics journal Visibility-based*, 1:477–493, 2000.
- [25] Ying Wang and Clarence W. de Silva. Multi-robot box-pushing single-agent q-learning vs team q-learning. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3694–3699, October 2006.
- [26] ZhiDong Wang and Vijay Kumar. Object closure and manipulation by multiple cooperating mobile robots. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 1:394–399, 2002.
- [27] Eric W. Weisstein. Piano mover's problem. <http://mathworld.wolfram.com/PianoMoversProblem.html>. Última vez consultada el 4 de Noviembre de 2008.
- [28] Seiji Yamada and Jun'ya Saito. Adaptive action selection without explicit communication for multi-robot box-pushing. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 31(3):398–404, 2001.
- [29] Atsushi Yamashita, Tamio Arai, Jun Ota, and Hajime Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on robotics and automation*, 19(2):223–237, Apr 2003.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL IPN

UNIDAD TAMAULIPAS

Cd. Victoria, Tamps. a 15 de diciembre de 2008

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará el C. Ezra Federico Parra González, declaramos que hemos revisado la tesis titulada:

“Desplazamiento de objetos por una comunidad de robots móviles, en un medio ambiente no estructurado”

Y consideramos que cumple con los requisitos para obtener el grado de Maestro en Ciencias en Computación.

Atentamente,

Dr. José Gabriel Ramírez Torres

Dr. José Torres Jiménez

Dr. Adriano de Luca Pennacchia



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000008917