



UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior  
DEPARTAMENTO DE ELECTRÓNICA



ESTRATEGIA DE ENRUTAMIENTO PARA LA MANIOBRA DE ENLACE  
A UN CONVOY DE VEHÍCULOS EN ENTORNOS URBANOS, ROBUSTA A  
LA INCERTIDUMBRE EN LOS TIEMPOS DE RECORRIDO

**TESIS DOCTORAL**

Fernando Valdés Villarrubia

2012



UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior  
DEPARTAMENTO DE ELECTRÓNICA

PROGRAMA DE DOCTORADO EN  
ELECTRÓNICA: CAPTACIÓN MULTISENSORIAL Y SISTEMAS ROBÓTICOS



ESTRATEGIA DE ENRUTAMIENTO PARA LA MANIOBRA DE ENLACE  
A UN CONVOY DE VEHÍCULOS EN ENTORNOS URBANOS, ROBUSTA A  
LA INCERTIDUMBRE EN LOS TIEMPOS DE RECORRIDO

Autor: Fernando Valdés Villarrubia

Directores: Dr. Felipe Espinosa Zapata  
Dr. Roberto Iglesias Rodríguez

2012

Tesis Doctoral



Dr. Felipe Espinosa Zapata, Profesor Titular de la Universidad de Alcalá, y

Dr. Roberto Iglesias Rodríguez, Profesor Titular de la Universidad de Santiago de Compostela

INFORMAN:

Que la Tesis Doctoral titulada “Estrategia de enrutamiento para la maniobra de enlace a un convoy de vehículos en entornos urbanos, robusta a la incertidumbre en los tiempos de recorrido”, presentada por D. Fernando Valdés Villarrubia, y realizada bajo nuestra dirección, dentro del campo de los sistemas inteligentes de transporte, reúne los méritos de calidad y originalidad para optar al Grado de Doctor.

Alcalá de Henares

Santiago de Compostela,

a 15 de junio de 2012.



Fdo. Felipe Espinosa Zapata



Fdo. Roberto Iglesias Rodríguez



Dr. Juan Jesús García Domínguez, Director del Departamento de Electrónica de la Universidad de Alcalá,

INFORMA:

Que la Tesis Doctoral titulada “Estrategia de enrutamiento para la maniobra de enlace a un convoy de vehículos en entornos urbanos, robusta a la incertidumbre en los tiempos de recorrido”, presentada por D. Fernando Valdés Villarrubia, y co-dirigida por Dr. Felipe Espinosa Zapata (Universidad de Alcalá) y Dr. Roberto Iglesias Rodríguez (Universidad de Santiago de Compostela), cumple con todos los requisitos científicos y metodológicos para ser defendida ante un Tribunal.

Alcalá de Henares, a 21 de junio de 2012.



Fdo. Juan Jesús García Domínguez.





# ABSTRACT

---

This thesis proposes an optimal routing strategy for intelligent transport units moving autonomously in an urban scenario. Inside this scenario, defined by a set of streets and intersections (nodes), there is a group of independent mobile units carrying out specific tasks. On the other hand there is a peripheral road along which a convoy of vehicles is moving continuously. This convoy is composed by one leader and several followers, without mechanic link between them. The task of the convoy is to travel along the peripheral route linking all those transport units that can be required to carry out some specific task. A transport unit of the convoy might have to leave it at any time to perform a task, and once the mission is finished, return to the convoy as quickly as possible.

This thesis focuses on providing solutions to the merging manoeuvre. In particular the return to the convoy of a single unit that has ended its mission. Taking into account that the movement of the convoy is restricted to the peripheral road, the meeting point amongst the convoy and any unit willing to return to it (pursuer), will be one of the peripheral nodes. On the other hand this meeting point must be selected in such a way that the link-up takes place in the shortest possible time, and the convoy always arrives at the meeting point after the pursuer.

The first objective consisted on solving the merging manoeuvre when the travelling times along every street are considered well known and deterministic. In this case the problem is divided into two stages: a) development of an algorithm able to determine the optimal meeting point where the pursuer will merge the convoy and, b) decision of the best route (analysing the minimum number of nodes) from the current pursuer's position to the meeting point. It is important to highlight that the full search process happens very quickly and can be applied in real time. This has special interest on complex environments.

The second goal consisted on extending our algorithms to a scenario where the times required to travel along the streets are not deterministic but random. This uncertainty is usual on real transport scenarios, and affects all the transport units. There are many sources of uncertainty, such as traffic density, weather conditions, hour of the day, etc. To model this uncertainty, we have used Gaussian probability distributions. This way the travelling times are now random variables characterized by a mean value and a variance. On the other hand, this non-deterministic behaviour prevents us from guaranteeing successful merging manoeuvres. Because of this, a novel parameter, named "Risk Factor", limits the failure probability of the merging manoeuvre. This factor also influences the solutions that our algorithms achieve: in particular the time required for the manoeuvres, and the number of trajectory re-plannings that are necessary to carry out while the pursuer approaches the meeting point.

The implementation of our proposal needs a remote centre with wireless connection with all transport units. The remote centre uses Dynamic Programming techniques, together with the information reported by the transport units, to keep updated relevant statistical values regarding the travelling times. The entire strategy was validated through different experiments, first on simulation and finally on real experiments. The Player/Stage tool allowed us to run simulated robots on a virtual map. The real experiments were carried out using P3-DX robots to perform the role of the leader and the pursuing unit.



# RESUMEN

---

Esta tesis propone una estrategia de enrutamiento óptima para unidades de transporte inteligente que se mueven de manera autónoma por un entorno urbano conocido. El entorno está definido por un conjunto de calles y cruces (nodos), y en su interior un grupo de unidades móviles independientes se encuentran realizando tareas específicas. Dicho entorno está rodeado por una ruta periférica por la que se mueve continuamente un convoy compuesto por un líder y un número determinado de unidades seguidoras, sin enlace mecánico entre ellos. La misión del convoy es concentrar las unidades independientes antes y después de que hayan realizado, de forma independiente, su tarea.

Básicamente, el trabajo se centra en dar solución a la maniobra de enlace consistente en lograr que la unidad independiente (perseguidora), partiendo de su ubicación actual en el interior del mapa, logre alcanzar el nodo periférico idóneo para unirse al convoy. Considerando que este último está limitado a circular por la ruta externa y por tanto no tiene acceso al interior del entorno, el enlace se realizará en uno de los nodos periféricos. El convoy sigue indefinidamente su trayectoria, por lo que la maniobra se considera exitosa siempre que la unidad independiente alcance el nodo de enlace antes que el convoy.

El primer objetivo es resolver la maniobra de enlace considerando conocidos los tiempos de recorrido entre los nodos del mapa. Objetivo que incluye dos fases: cálculo del nodo óptimo de enlace y de la ruta que lleve a la unidad perseguidora hasta el mismo. Se entiende por nodo óptimo de enlace aquél que garantiza un tiempo mínimo de maniobra. Además, se ha diseñado un algoritmo de enrutamiento que explora el menor número de nodos posibles lo que garantiza su eficiencia computacional y su idoneidad para su ejecución en tiempo real, de especial interés en entornos complejos.

El segundo objetivo es extender estos algoritmos a un entorno donde los tiempos de recorrido entre nodos no son conocidos. Esta incertidumbre, inherente a los tiempos de recorrido de todas las unidades, es propia de escenarios de transporte reales y tiene su origen en diversas fuentes como densidad variable de tráfico, condiciones meteorológicas, momento del día, etc. Para caracterizarla se ha propuesto un modelo gaussiano, donde los tiempos de recorrido son tratados como variables aleatorias parametrizadas por su valor medio y varianza. Por otra parte, este comportamiento no determinista impide garantizar de forma absoluta el éxito seguridad la maniobra de enlace. Por ello, se introduce el parámetro de diseño “Factor de Riesgo”, que limita la probabilidad de fallo de la maniobra de enlace. Este factor condiciona además el tiempo de maniobra y el número de re-planificaciones intermedias hasta llegar al nodo final.

En la solución propuesta se incluye un centro remoto al que están conectadas de forma inalámbrica todas las unidades de transporte. En el centro remoto se registran los tiempos de recorrido entre nodos consecutivos proporcionados por las unidades de transporte y se estiman los parámetros estadísticos temporales entre nodos no consecutivos mediante técnicas recursivas de Programación Dinámica.

Finalmente, se ha procedido a la validación experimental de la propuesta global. En una primera fase se ha recurrido a la herramienta Player/Stage para validar mediante simulación los cálculos desarrollados a partir de un mapa diseñado al efecto. Superada esta, se ha utilizado un demostrador real donde la función de unidad líder y unidad perseguidora ha sido desarrollada por robots Pioneer P3-DX.



*A mis padres*



# AGRADECIMIENTOS

---

Tras años de trabajo son muchas las personas que de manera directa o indirecta, han aportado su experiencia, consejos y buenos deseos para la culminación de esta investigación. A todos ellos quiero expresar hoy mi agradecimiento, esperando no olvidar a nadie y si es así de todos modos: muchas gracias.

En el ámbito académico quiero agradecer primeramente a mis directores de tesis:

Al Dr. Felipe Espinosa Zapata, por su experiencia, infinita paciencia e inestimable ayuda durante todos estos años.

Al Dr. Roberto Iglesias Rodríguez, mi segundo director, por su apoyo, consejos y toda la ayuda prestada para la realización de este trabajo.

También quisiera agradecer al Departamento de Electrónica de la Universidad de Alcalá, gracias al cual fue posible alcanzar esta meta. En particular, muchas gracias al Dr. José Luis Lázaro y al Dr. Manuel Mazo por su apoyo y amistad.

Por supuesto quiero agradecer también a mis compañeros de laboratorio, cubanos y españoles, que a lo largo de estos años han compartido experiencias, alegrías y preocupaciones conmigo. A Marcelo Susel, Isabel, Carlos, Cano, Pedrito, Katia, Peña. Muchas gracias a todos.

A Luna y Haidee por sus consejos y preocupaciones, y buenos ratos compartidos.

Del otro lado del Atlántico, quisiera agradecer a los profesores Luis Lamezón y Ricardo Núñez, de La Universidad de Oriente, que nunca dudaron en apoyarme en esta empresa.

También quiero dar gracias a mis amigos de toda la vida, a Jesús, Fiffe, Daniel, Marcos, Charro, por estar ahí para lo bueno, para lo malo, y para lo peor.

A mi familia en Cuba, en especial a mis padres por apoyarme siempre, estando a su lado y estando tan lejos.

A Yisel, por darme tanto y pedir tan poco, por aguantarme, por todo su amor y por estar siempre junto a mí.

A mi familia adoptiva de Málaga, por su ánimo en esta recta final.

Finalmente, terminaré olvidando a algunos. A todos ellos, pido disculpas y doy las gracias.





# ÍNDICE

---

<b>1. INTRODUCCIÓN</b>	<b>1</b>
<b>1.1. Sistemas de Transporte Inteligente. Retos del guiado cooperativo</b>	<b>1</b>
<b>1.2. Contexto de la tesis y motivación</b>	<b>5</b>
<b>1.3. Objetivos de la tesis y contribuciones</b>	<b>7</b>
<b>1.4. Estructura de la tesis</b>	<b>8</b>
<b>2. ESTADO DEL CONOCIMIENTO</b>	<b>9</b>
<b>2.1. Revisión de algoritmos de enrutamiento</b>	<b>9</b>
2.1.1. Estáticos y dinámicos	12
2.1.2. Algoritmos de enrutamiento en tiempo real	20
2.1.3. Enrutamiento con incertidumbre en los tiempos de recorrido	21
<b>2.2. Herramientas comerciales para el cálculo de rutas</b>	<b>23</b>
<b>3. ENRUTAMIENTO ÓPTIMO Y EFICIENTE PARA MANIOBRAS DE ENLACE</b>	<b>25</b>
<b>3.1. Planteamiento del problema</b>	<b>25</b>
3.1.1. El escenario de transporte	26
3.1.2. El Centro Remoto	29
3.1.3. Las unidades de transporte	30
3.1.3.1. Tareas llevadas a cabo por el convoy	30
3.1.3.2. Tareas correspondientes a la unidad perseguidora	31
<b>3.2. Resolución del problema de enrutamiento</b>	<b>31</b>
3.2.1. Terminología y definiciones	31
3.2.2. Nodo óptimo de enlace	34
3.2.2.1. Cálculo del tiempo entre nodos aplicando Programación Dinámica	35
3.2.2.2. Cálculo del nodo óptimo de enlace	38
3.2.3. Algoritmo eficiente para cálculo de ruta de enlace	38
3.2.3.1. Cálculo de la ruta de enlace	39
3.2.3.2. Eficiencia y optimalidad de la solución planteada	40
<b>3.3. Resultados de simulación</b>	<b>41</b>
<b>4. EFECTO DE LA INCERTIDUMBRE EN LOS TIEMPOS DE RECORRIDO</b>	<b>47</b>
<b>4.1. Introducción</b>	<b>47</b>
<b>4.2. Modelado de la incertidumbre en los tiempos de recorrido</b>	<b>48</b>
4.2.1. Actualización de parámetros de incertidumbre	50

4.2.1.1. Actualización de la media y varianza de los tiempos de recorrido entre nodos adyacentes	50
4.2.1.2. Actualización de la media y varianza de los tiempos de recorrido entre nodos no adyacentes	51
<b>4.3. Factor de Riesgo</b>	<b>52</b>
<b>4.4. Evaluación dinámica de la maniobra de enlace</b>	<b>55</b>
4.4.1. Cálculo del nodo de enlace en función del riesgo asumido	56
4.4.2. Re-evaluación de la ruta de enlace y re-planificación	56
<b>4.5. Resultados de simulación</b>	<b>58</b>
4.5.1. Prueba 1. Maniobra de enlace con factor de riesgo del 5 %	60
4.5.2. Prueba 2. Maniobra de enlace con factor de riesgo del 50 %	61
4.5.3. Prueba 3. Maniobra de enlace con factor de riesgo del 95 %	62
<b>4.6. Alternativas para el modelado de incertidumbre en los tiempos de recorrido</b>	<b>64</b>
4.6.1. Función de distribución Log-normal	65
4.6.2. Función de distribución Gamma	66
4.6.3. Función de distribución Exponencial	67
<b>5. DEMOSTRADOR CON ROBOTS PIONEER P3-DX</b>	<b>69</b>
<b>5.1. Introducción.</b>	<b>69</b>
<b>5.2. Descripción del demostrador</b>	<b>70</b>
5.2.1. Elementos	70
5.2.2. Entorno	74
<b>5.3. Resultados experimentales</b>	<b>74</b>
5.3.1. Ensayo experimental con factor de riesgo del 0 %	75
5.3.2. Ensayo experimental con factor de riesgo del 100 %	76
<b>6. CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>79</b>
<b>6.1. Conclusiones</b>	<b>79</b>
<b>6.2. Líneas futuras de investigación</b>	<b>81</b>
<b>ANEXO 1</b>	<b>85</b>
<b>BIBLIOGRAFÍA</b>	<b>89</b>

# ÍNDICE DE FIGURAS

---

<i>Figura 1.1. Ejemplo de congestión vial (izda) y uno de sus graves efectos: la polución atmosférica (dcha).</i> .....	1
<i>Figura 1.2. Demostración de vehículos en convoy realizada por PATH, en San Diego, California.</i> .....	5
<i>Figura 1.3. Escena de animación mostrando la conducción en formación en convoy del Proyecto SARTRE.</i> .....	5
<i>Figura 2.1. Grafo dirigido (izda) y su árbol correspondiente (dcha), comenzando la expansión en el nodo A.</i> .....	10
<i>Figura 2.2. Grafo de la Figura 2.1 representado con listas y matriz de adyacencia, respectivamente</i> .....	11
<i>Figura 2.3. Algoritmo Primero en Anchura (BFS). Los nodos son enumerados según el orden de análisis.</i> .....	13
<i>Figura 2.4. Algoritmo Primero en Profundidad (DFS)</i> .....	14
<i>Figura 2.5. Algoritmo de búsqueda bi-direccional, a partir del origen (izda) y a partir del destino (dcha).</i> .....	15
<i>Figura 3.1. Esquema básico de las partes que componen la estrategia de enrutamiento.</i> .....	25
<i>Figura 3.2. Vista parcial del Casco Histórico de Alcalá de Henares. (izda) Mapa métrico. (dcha) Mapa topológico correspondiente al mapa métrico, con las zonas interior y exterior definidas. Los números que aparecen en algunos de los arcos representan el tiempo requerido (en segundos) por un vehículo para atravesar las calles representadas por estos arcos, a una velocidad de recorrido de 24 Km/h.</i> .....	27
<i>Figura 3.3. Mapa topológico de la Figura 3.2 en el que cada tramo simple tiene asociado un valor (de forma cualitativa: bajo, medio o alto) de la velocidad con que una unidad móvil lo recorrerá.</i> .....	28
<i>Figura 3.4. Ejemplo de mapa topológico simplificado de un escenario de transporte. Se muestra la ubicación inicial de la unidad perseguidora y la ruta cíclica seguida por el convoy.</i> .....	32
<i>Figura 3.5. Ejemplo de tramos simples entre nodos adyacentes <math>(n_i - n^h, s_i - s_{i+1})</math>, o tramos combinados entre nodos no adyacentes <math>(n_i - s_j, s_i - s_j)</math>.</i> .....	33
<i>Figura 3.6. Diagrama de bloques general de la estrategia de enrutamiento, con la mayoría de funciones de la unidad perseguidora y el Centro Remoto.</i> .....	34
<i>Figura 3.7. Algoritmo para calcular los tiempos de recorrido estimados entre nodos periféricos no adyacentes.</i> .....	37
<i>Figura 3.8. Algoritmo para calcular los tiempos de recorrido estimados entre un nodo interior y uno periférico.</i> .....	37
<i>Figura 3.9. Algoritmo de enrutamiento eficiente para obtener la ruta de enlace entre la ubicación actual de la unidad perseguidora y el nodo <math>s_e</math> previamente obtenido</i> .....	40

<i>Figura 3.10. Mapa topológico correspondiente a un entorno urbano imaginario. Los números asociados a cada tramo representan los tiempos estimados entre nodos adyacentes.....</i>	<i>42</i>
<i>Figura 3.11. Ejemplo de dos soluciones diferentes para las combinaciones <math>n^0 = n_9</math>, <math>s^0 = s_4</math> y <math>n^0 = n_9</math>, <math>s^0 = s_7</math>. La línea discontinua muestra el punto de partida y la ruta seguidapor el convoy hasta el nodo de enlace, mientras la línea continua corresponde a la ruta de la unidad perseguidora.....</i>	<i>43</i>
<i>Figura 3.12. Mapa empleado en el Ejemplo 2, construido sobre la plataforma de simulación Player/Stage.....</i>	<i>44</i>
<i>Figura 4.1. Función de densidad de probabilidad de la variable aleatoria gaussiana <math>x</math> .....</i>	<i>48</i>
<i>Figura 4.2. Función de densidad de probabilidad de los tiempos de recorrido de unidad perseguidora <math>t_p</math> y convoy <math>t_c</math> .....</i>	<i>49</i>
<i>Figura 4.3. Algoritmo para calcular media y varianza de los tiempos de recorrido estimados entre nodos periféricos no adyacentes.....</i>	<i>51</i>
<i>Figura 4.4. Algoritmo para calcular la media y varianza de los tiempos de recorrido estimados entre cada nodo interior y cada nodo periférico.....</i>	<i>52</i>
<i>Figura 4.5. Función de densidad mostrando la probabilidad de que se cumpla: <math>t_p \leq t_p^*</math> .....</i>	<i>53</i>
<i>Figura 4.6. Representación gráfica de <math>t_p^*</math> para distintos valores de <math>z_p(\theta)</math>: 2.33, 0 y -2.33. ....</i>	<i>54</i>
<i>Figura 4.7. Solapamiento de las funciones de densidad de probabilidad de perseguidor y convoy para diferentes casos del riesgo asumido en el cálculo de la maniobra de enlace. ....</i>	<i>55</i>
<i>Figura 4.8. Diagrama de bloques de la estrategia de enrutamiento contemplando incertidumbre en los tiempos de recorrido. ....</i>	<i>57</i>
<i>Figura 4.9. Cálculo de <math>T(s_x, s_e)</math> cuando el convoy se encuentra entre dos nodos periféricos.....</i>	<i>57</i>
<i>Figura 4.10. Distribución de medias y varianzas de los tramos para diferentes zonas del mapa bajo estudio. ....</i>	<i>59</i>
<i>Figura 4.11. Imágenes tomadas durante las dos simulaciones de Player/Stage donde ambas maniobras de enlace tienen el mismo riesgo del 50 %. (A) muestra una maniobra exitosa, mientras la mostrada en (B) es fallida. ....</i>	<i>62</i>
<i>Figura 4.12. Número de re-planificaciones realizadas por la unidad perseguidora en las tres pruebas. ....</i>	<i>64</i>
<i>Figura 4.13. Número de veces que cada nodo periférico <math>s_e</math> es elegido como nodo de enlace durante las simulaciones realizadas en las Pruebas 1 (riesgo del 5 %), 2 (riesgo del 50 %) y 3 (riesgo del 95 %). ....</i>	<i>64</i>
<i>Figura 5.1. Diagrama con los componentes del demostrador.....</i>	<i>70</i>
<i>Figura 5.2. Centro Remoto y vista parcial del entorno diseñado para la demostración. ....</i>	<i>71</i>
<i>Figura 5.3. Plataforma robótica Pioneer P3-DX y algunas de sus posibles aplicaciones [P3-DX, 2012].....</i>	<i>71</i>

<i>Figura 5.4. Componentes añadidos al Pioneer P3-DX para la realización de experimentos.....</i>	<i>72</i>
<i>Figura 5.5. Plataforma básica (izda) y mejorada (dcha) en el grupo de investigación de robot P3-DX. ....</i>	<i>73</i>
<i>Figura 5.6. Arquitectura de comunicaciones de las unidades móviles con el CR.....</i>	<i>73</i>
<i>Figura 5.7. A la izquierda, representación gráfica del escenario del experimento, con las tres áreas claramente diferenciadas por el valor medio y varianza de la velocidad a la cual pueden moverse las unidades robóticas. A la derecha, una vista parcial del escenario real empleado. Uno de los robots hace las veces de unidad perseguidora mientras el otro asume el rol de líder del convoy.....</i>	<i>74</i>
<i>Figura 5.8. Demostración de la estrategia de enrutamiento implementada en robots Pioneer P3-DX, aplicando un Factor de Riesgo del 0 % (<math>\delta = 0</math>). A la izquierda, las ubicaciones simultáneas de ambas unidades y el nodo de enlace calculado en cada momento. A la derecha, varias instantáneas tomadas durante el experimento, correspondientes a los mismos instantes.....</i>	<i>75</i>
<i>Figura 5.9. Resultados obtenidos por la estrategia de enrutamiento aplicando un Factor de Riesgo del 100 % (<math>\delta = 1</math>). A la izquierda se representan las ubicaciones de ambas unidades robóticas a diferentes instantes de tiempo, así como el nodo de enlace elegido en cada momento. A la derecha pueden apreciarse instantáneas tomadas durante el experimento, correspondientes a los mismos instantes de tiempo.....</i>	<i>77</i>



# ÍNDICE DE TABLAS

---

TABLA 3.1. TIEMPOS ESTIMADOS DE RECORRIDO $T(s_i, s_j)$ ENTRE CADA PAR DE NODOS PERIFÉRICOS ALCANZABLES DEL MAPA MOSTRADO EN FIGURA 3.10.....	42
TABLA 3.2. TIEMPOS ESTIMADOS DE RECORRIDO $T(n_i, s_j)$ ENTRE CADA POSIBLE PAR DE NODOS INTERIOR-PERIFÉRICO ALCANZABLE DEL MAPA MOSTRADO EN FIGURA 3.10.....	43
TABLA 3.3. NODO DE ENLACE Y RUTA OBTENIDA PARA DIFERENTES PARES $(n^0, s^0)$ .....	43
TABLA 3.4. COMPARACIÓN DE NODOS EXPANDIDOS Y TIEMPOS DE EJECUCIÓN DEL ALGORITMO PROPUESTO Y DEL ALGORITMO DE REFERENCIA A*, EN EL CÁLCULO DE LAS RUTAS PARA DIFERENTES PARES $(n^0, s^0)$ .....	45
TABLA 4.1. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 1, ASUMIENDO UN RIESGO DE FALLO DEL 5 %.....	60
TABLA 4.2. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 2, ASUMIENDO UN RIESGO DE FALLO DEL 50 %.....	61
TABLA 4.3. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 3, ASUMIENDO UN RIESGO DE FALLO DEL 95 %.....	63





# GLOSARIO

---

## Glosario de siglas y acrónimos

<i>ABS</i>	Anti-lock Braking System
<i>ADAS</i>	Advanced Driver Assistance System
<i>BFS</i>	Breadth-First Search algorithm
<i>CC</i>	Cruise Control system
<i>CR</i>	Centro Remoto
<i>D*</i>	Dynamic A*
<i>DFS</i>	Depth-First Search Algorithm
<i>E-call</i>	Emergency Call
<i>ESP</i>	Electronic Stability Control
<i>ETC</i>	Electronic Toll Collection
<i>FALCONS</i>	FAst Learning and CONverging Search
<i>GIS</i>	Geographic Information System
<i>GPS</i>	Global Positioning System
<i>IDA*</i>	Iterative-Deepening A* Algorithm
<i>ITS</i>	Intelligent Transportation Systems
<i>LPA*</i>	Lifelong Planning A* Algorithm
<i>LRTA*</i>	Learning Real-Time A*
<i>PD</i>	Programación Dinámica
<i>R-SPP</i>	Restricted Shortest Path Problem
<i>RTA*</i>	Real-Time A*
<i>RTD*</i>	Real Time D*
<i>SPP</i>	Shortest Path Problem
<i>SP-SPP</i>	Single Pair Shortest Path Problem
<i>SS-SPP</i>	Single Source Shortest Path Problem
<i>STSPTW</i>	Stochastic Traveling Salesman Problem with Time Windows
<i>TD-SSP</i>	Time Depend Shortest Path Problem
<i>V2I</i>	Vehicle – Infrastructure communications
<i>V2V</i>	Vehicle – Vehicle communications
<i>VIA-T</i>	Sistema de telepeaje en autopistas españolas
<i>VRP</i>	Vehicle Routing Problem



# 1

## INTRODUCCIÓN

La Tesis se enmarca dentro de los Sistemas de Transporte Inteligente, y particularmente dentro de los llamados sistemas cooperativos. Esto supone que los vehículos tienen capacidad de interacción entre ellos y con la vía, facilitando así las tareas de supervisión, control y guiado, entre otras. En este capítulo se plantea el contexto, motivación, objetivos a alcanzar y contribuciones resultantes del trabajo de investigación.

### 1.1. Sistemas de Transporte Inteligente. Retos del guiado cooperativo

El crecimiento de la población mundial y la consiguiente urbanización de grandes espacios, ha provocado un incremento notable del tráfico rodado (véase Figura 1.1.izda) y de la demanda de infraestructuras urbanas, principalmente en los países desarrollados o en vías de desarrollo. Esto ha traído como consecuencia un aumento de la congestión vial (especialmente en grandes ciudades) y de importantes demoras en las carreteras y autopistas de acceso a las mismas, donde en hora punta el tráfico se ve colapsado cotidianamente. Estos elementos, junto a otros igual o más críticos como la polución del aire (véase Figura 1.1.dcha) y los elevados niveles de ruido [Houghton *et al.*, 2009], provocan serias consecuencias en la salud y bienestar de la población e implican altos costes materiales para cualquier país afectado [Levy *et al.*, 2010].



Figura 1.1. Ejemplo de congestión vial (izda) y uno de sus graves efectos: la polución atmosférica (dcha).

Por tanto, es imprescindible contar con estos elementos al evaluar cualquier política seria de planificación urbana, por lo que el desarrollo y explotación de estrategias avanzadas de gestión y ayuda al tráfico se ha convertido en los últimos años en una meta apremiante a nivel mundial, con el fin de lograr ciudades menos congestionadas, más limpias y seguras, y por tanto con un mayor bienestar para sus habitantes.

Los Sistemas de Transporte Inteligente (*ITS*, por sus siglas en inglés), tienen por objetivo aportar soluciones tecnológicas que ayuden a paliar los efectos negativos del transporte a gran escala. Estos se apoyan en ingenierías electrónica, informática y de comunicaciones y, aunque presentes en áreas como el transporte ferroviario y aéreo, están fundamentalmente orientados a brindar soluciones al transporte rodado y la infraestructura de carretera, teniendo como meta mejorar la operatividad y sostenibilidad de los mismos, incrementando su eficiencia, y mejorando la seguridad y confort de quienes lo utilizan.

Los *ITS* van de la mano con las tecnologías de vanguardia y abarcan un amplio campo de acciones, tanto para entornos interurbanos (autopistas y autovías), como para espacios urbanos. Entre las más extendidas podemos mencionar:

- Los sistemas avanzados de asistencia a la conducción (*ADAS*), integrados en el propio vehículo y orientados a evitar posibles accidentes de tráfico o mitigar sus efectos. El sistema antibloqueo de frenos (*ABS*), el control de velocidad de cruceo (*CC*) o de estabilidad (*ESP*), son algunos ejemplos.
- Sistemas de navegación mediante *GPS*, que permite conocer la ubicación del vehículo en todo momento, y planificar trayectorias a un destino deseado.
- Sistemas de cobro electrónico de peaje en autopistas (*ETC*) o tele-peaje. El pago del peaje se realiza a través de una comunicación inalámbrica entre el vehículo y la caseta de cobro, no siendo necesario que los primeros se detengan completamente ante la barrera y generen congestión en las zonas de peaje. En España, el tele-peaje emplea el sistema *VIA-T* [VIA-T, 2011], implementado en la gran mayoría de autopistas españolas.
- Paneles electrónicos de información. Señales visuales situadas sobre la calzada que brindan una valiosa información de tráfico y viaje a los usuarios de la vía, actualizada constantemente.
- Gestión de flotas en el transporte de mercancía. Principalmente enfocado a la reducción de costes de operación, desarrollando soluciones para la localización y seguimiento en tiempo real de vehículos de carga, la gestión de rutas y puntos de distribución, o la supervisión del estado de las carreteras, entre otras. Empresas como *GMV* y *Zenithal* se encargan de brindar este tipo de servicios en España [GMV, 2010], [Zenital, 2011].

A raíz de la notable expansión que en los últimos 10 años ha mostrado el mundo de las comunicaciones (telefonía móvil, *GPS*, internet, etc.), un campo que paralelamente ha ido en ascenso dentro de los *ITS* es el de los sistemas cooperativos. La meta de este relativamente

nuevo concepto es dotar a todos los vehículos con capacidad de cooperación, no solo manteniendo una interacción coordinada y constante con la infraestructura del entorno (comunicación vehículo-infraestructura -V2I-), sino también con el resto de vehículos de la vecindad (comunicación vehículo-vehículo -V2V-), compartiendo información relevante acerca de la ubicación y estado de cada vehículo próximo y de la propia vía. Así, se pretende que los usuarios dispongan de información de calidad acerca de su entorno lo que redundaría en una mejora significativa de la conducción, posibilitando una circulación más fluida y segura. Esta idea no es sino un paso más hacia un escenario futuro en el que los vehículos cuenten además con capacidad de auto-maniobra, formando un sistema totalmente coordinado en el que la intervención humana en la conducción sea opcional.

La comunicación V2I está principalmente orientada a incrementar la eficiencia de la conducción, evitar atascos y accidentes, y, en definitiva, aprovechar al máximo las potencialidades de la infraestructura, todo ello sin dejar de lado la seguridad de los usuarios. Entre sus principales aplicaciones están la utilización de paneles de señalización variable para brindar información de tráfico en tiempo real; el aviso de posibles incidentes u obstáculos en la ruta, posibilitando al conductor el cambio de itinerario con suficiente antelación [CoVel Project, 2011]; o el control por exceso de velocidad, identificando la máxima velocidad permitida en cada sección de la vía y alertando al vehículo si la excede, llegando incluso a limitar su velocidad de manera automática [Abbas *et al.*, 2010].

En este sentido, en Europa, proyectos como *SAFESPOT* (Smart Vehicle on Smart Roads) [SAFESPOT Project, 2011] o *COOPERS* (Co-operative Systems for Intelligent Road Safety) [COOPERS Project, 2011] se empeñan en mejorar la seguridad vial, mediante la incorporación de técnicas que ayudan a prevenir accidentes de tráfico. Para ello se han desarrollado plataformas de comunicaciones en vehículos cooperativos en la que tanto la vía como el propio vehículo constituyen fuentes de información, dedicadas a mejorar la percepción de los conductores acerca de lo que les rodea, detectar situaciones críticas con antelación y ayudarlos a realizar las maniobras adecuadas en cada situación.

Otra iniciativa muy interesante es la llamada de emergencia automática (*E-call*), impulsada por la Comisión Europea con el objetivo de brindar ayuda inmediata ante la ocurrencia de accidentes de tráfico. Ante un siniestro, un dispositivo electrónico instalado en el vehículo colisionado emite una llamada automática de emergencia al 112, conteniendo los datos del mismo y fundamentalmente su ubicación. Esto posibilitaría una reducción de entre un 40 a 50 por ciento del tiempo de respuesta de los servicios de urgencia, estimándose que podrían salvarse hasta un 15 por ciento más de vidas en caso de siniestros [EIS, 2012].

Por otra parte, respecto a las comunicaciones V2V, el trabajo está más orientado a mejorar la seguridad en la conducción. Mediante una constante comunicación entre vehículos y apoyados en localización *GPS* o tecnología radar, es posible monitorizar la distancia de seguridad con los vehículos circundantes, lanzar un aviso si se reduce significativamente o ante el peligro inminente de colisiones entre dos o más vehículos [Ashtankar *et al.*, 2009], [Toor *et al.*, 2008], [Zhu *et al.*, 2010], [RITA, 2011].

En Europa, el proyecto *CVIS (Cooperative Vehicle-Infrastructure Systems)* [CVIS Project, 2011], coordinado por *ERTICO-ITS Europa*, está orientado al diseño y desarrollo de tecnologías que permitan la interacción continua entre vehículos y entre estos y la infraestructura. Entre sus objetivos está el crear soluciones técnicas que posibiliten a todos los vehículos comunicarse entre sí y con la infraestructura a través de una arquitectura de comunicación multicanal, y potenciar protocolos de cooperación entre las distintas partes del sistema.

En territorio español, el proyecto *MARTA (Movilidad y Automoción con Redes de Transporte Avanzadas)* [MARTA, 2011], enmarcado en el programa *CENIT* de la iniciativa *INGENIO 2010*, cuenta con la participación de varios centros universitarios y empresas españolas de diversos sectores y tiene como objetivo la investigación y desarrollo de las comunicaciones vehículo-vehículo y vehículo-infraestructura vial, para así contar con soluciones tecnológicas fiables y seguras que gestionen el tráfico de un modo más eficiente, reducir el número de accidentes y disminuir el impacto ambiental de la automoción.

Todo lo anterior pone de manifiesto que la cooperación entre unidades es una de las líneas de investigación dentro de los *ITS* para mitigar los problemas ya comentados inherentes al tráfico rodado. Como es sabido, la formación en convoy es una de las alternativas a considerar en el ámbito del transporte [California PATH, 2011], [SARTRE, 2011].

Un convoy está formado por un vehículo líder y dos o más seguidores, sin conexión física entre ellos. En convoyes automatizados el líder puede ser conducido en modo manual o automático, mientras que los seguidores implementan los sistemas de sensado, comunicación y control necesarios para garantizar la estabilidad (lateral y longitudinal) del conjunto o estabilidad de cadena [Espinosa *et al.*, 2007]. Además del seguimiento manteniendo una distancia de seguridad, la formación y disgregación del conjunto son objetivos abiertos a discusión [Valdés *et al.*, 2006].

*PATH (Partners for Advanced Transit and Highways)* [California PATH, 2011], es un programa multidisciplinario administrado por el *Institute of Transportation Studies* de la Universidad de California. *PATH* aglutina a un conjunto de proyectos de investigación enfocados principalmente en temas de seguridad del transporte y operaciones de tráfico y tiene como misión desarrollar soluciones a los problemas del transporte por carretera a través de la investigación en estos campos y en colaboración con expertos en tecnología de la información, ingeniería eléctrica, economía y otros. Uno de sus campos de investigación está dedicado a la conducción en convoy mostrando resultados al respecto, con demostraciones controladas de convoyes de vehículos en autopistas convencionales, como se muestra en la Figura 1.2.



Figura 1.2. Demostración de vehículos en convoy realizada por PATH, en San Diego, California.

Por su parte, *SARTRE (Safe Road Trains for the Environment)* [SARTRE, 2011], financiado por la Comisión Europea bajo el 7º programa marco, tiene como objetivo principal fomentar un cambio en el uso del transporte privado por carretera, a través del empleo de convoyes de vehículos guiados de forma automática, desarrollando estrategias y tecnologías que les permitan operar en autopistas convencionales. El sistema a desarrollar estará formado por un vehículo líder controlado por un conductor profesional. Los vehículos que pretendan formar parte del convoy entrarán en un modo de control semi-automático que permitirá a sus conductores dedicarse a otras actividades, como leer o hablar por teléfono, mientras el vehículo permanezca acoplado al convoy.

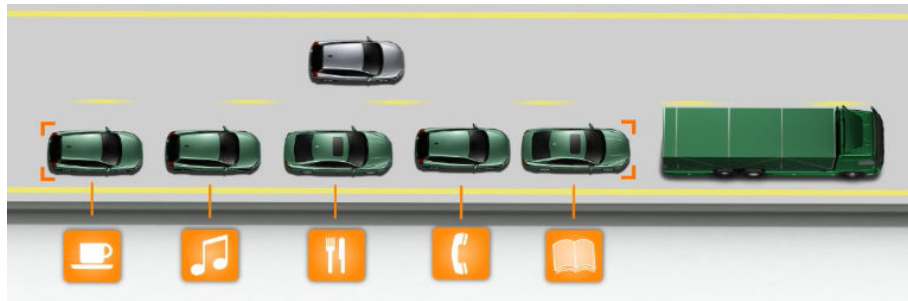


Figura 1.3. Escena de animación mostrando la conducción en formación en convoy del Proyecto SARTRE.

## 1.2. Contexto de la tesis y motivación

Uno de las problemáticas a tratar en el campo del guiado en convoy es la referente a la aportación de estrategias de enrutamiento para la realización de maniobras de enlace con el convoy desde un punto remoto y separación del convoy hasta un punto remoto. En el grupo de investigación *GEINTRA*, del Departamento de Electrónica de la Universidad de Alcalá, se ha estado trabajando en esta línea, fundamentalmente a raíz del proyecto *COVE (Sistema de Transporte Inteligente para Guiado Cooperativo de Vehículos Eléctricos en Entornos Especiales)* [COVE, 2009].

Del proyecto *COVE* resulta un demostrador de varios robots (prototipos de vehículos eléctricos) circulando de forma autónoma en convoy, donde el líder recibe las consignas de



recorrido desde un centro remoto. Entre los objetivos alcanzados en el proyecto cabe destacar:

- Diseño, simulación e implementación de soluciones para la formación del convoy y el seguimiento estable del mismo [Rodríguez *et al.*, 2006], [Espinosa *et al.*, 2007], [Espinosa *et al.*, 2008a].
- Gestión de la entrada y salida de unidades del convoy, incluyendo las correspondientes trayectorias con un extremo estático y el otro dinámico [Valdés *et al.*, 2006], [Rodríguez *et al.*, 2009].
- Diseño de un sistema inteligente para la toma de decisiones sobre la ruta más adecuada, desarrollando aplicaciones que ayudan a definir maniobrabilidad de unidades de un convoy [Valdés *et al.*, 2011b], [Valdés *et al.*, 2011c].
- Estudio del efecto de las emisiones electromagnéticas de vehículos eléctricos tanto en el comportamiento del mismo como en la interacción con otros del convoy [De Santiago *et al.*, 2010].
- Construcción de un demostrador, incluyendo la integración de sistemas sensoriales, de control y comunicación inalámbrica para el guiado cooperativo de las unidades del convoy [Santos *et al.*, 2010], [Salazar, 2011], [Espinosa *et al.*, 2011].

Esta tesis se plantea para aportar soluciones a parte de los objetivos incluidos en el Proyecto COVE (2006-2009), y tiene su continuidad con el Proyecto VISNU (*Identification and Interaction of Multiple Agents in Intelligent Spaces Using a Camera Array*, 2010-12) [VISNU, 2011]. Dentro del proyecto VISNU se han abordado tareas relacionadas con el guiado de robots móviles, generación de trayectorias y definición de algoritmos de enrutamiento.

Su principal motivación está en la contribución a la maniobrabilidad de vehículos autónomos circulando en formación de convoy en escenarios especiales de transporte urbano. Como entornos especiales se entienden aquellos que cumplen las siguientes características: áreas donde el tráfico convencional está o bien restringido o bien prohibido, las velocidades de circulación son reducidas (no superan los 50 Km/h) y las trayectorias suelen presentar fuertes no linealidades. Ejemplo de este tipo de escenarios son: núcleos de interés histórico-cultural, parques temáticos o de ocio, entornos portuarios o aeroportuarios, núcleos residenciales, campus universitarios, etc.

Teniendo en cuenta las características de este tipo de entorno, el sistema propuesto da solución al problema de una unidad móvil que, una vez realizada la tarea encomendada como unidad independiente, ha de enlazar con el convoy circulando por la periferia del escenario de transporte.

### 1.3. Objetivos de la tesis y contribuciones

El objetivo general de la tesis consiste en crear una estrategia eficiente de enrutamiento que garantice a una unidad móvil unirse a un convoy en movimiento. El criterio de búsqueda es minimizar el tiempo empleado hasta realizar la maniobra de enlace, así como el tiempo de espera del convoy en el nodo elegido de la periferia. Todo ello considerando las incertidumbres inherentes a los tiempos de recorrido tanto de la unidad seguidora como del convoy.

Para lograrlo, se han definido los siguientes objetivos particulares:

- Calcular el nodo óptimo de enlace. Este debe ser aquel que garantice una incorporación de la unidad seguidora al convoy en un tiempo mínimo. La maniobra se realiza con éxito si el convoy alcanza dicho nodo después de la unidad perseguidora.
- En una primera aproximación, con tiempos conocidos de recorrido entre nodos adyacentes, diseño de un algoritmo de enrutamiento eficiente desde el punto de vista computacional para calcular la trayectoria óptima (de menor tiempo invertido) desde el punto de partida del perseguidor hasta el nodo de enlace.
- Rediseño de la solución considerando las incertidumbres inherentes a los tiempos de recorrido de cada tramo del mapa en los cálculos del nodo y ruta de enlace. Se asume un comportamiento gaussiano para modelar dicha incertidumbre. La aleatoriedad del nuevo enfoque obliga a re-evaluar la solución prevista tras la consecución de cada nodo intermedio.
- Incorporación de un parámetro que acote la probabilidad de encontrar una solución válida: factor de riesgo. Un mayor factor de riesgo conduce a soluciones más rápidas pero con menor probabilidad de éxito.
- Validación de las propuestas en un entorno simulado utilizando la herramienta Player/Stage, así como en un demostrador real empleando unidades robóticas Pioneer 3-DX.

Del cumplimiento de los objetivos propuestos, se derivaron los siguientes resultados:

- Aplicación de técnicas recursivas basadas en Programación Dinámica (*PD*) al cálculo del nodo óptimo de enlace [Valdés *et al.*, 2009], [Valdés *et al.*, 2011b].
- Algoritmo de enrutamiento eficiente capaz de obtener la ruta de enlace evaluando el mínimo número de nodos posibles, haciéndolo idóneo para aplicaciones de tiempo real [Valdés *et al.*, 2011a], [Valdés *et al.*, 2011b].
- Mecanismo de actualización de los parámetros estadísticos asociados a cada tramo interno y periférico. Cada vez que una unidad cualquiera concluye un tramo, se

registra el tiempo empleado en hacerlo y se actualiza el valor histórico del tiempo de recorrido de dicho tramo. De esta forma es posible contar con la información más reciente en la toma de decisiones [Valdés *et al.*, 2011b], [Valdés *et al.*, 2011c].

- Desarrollo de una estrategia de enrutamiento robusta por considerar las incertidumbres de los tiempos de recorrido, y adaptativa por tener en cuenta los nuevos tiempos de recorrido aportados on-line por el resto de unidades del escenario de transporte [Valdés *et al.*, 2012a].
- Introducción de un factor de riesgo que acota la probabilidad de éxito/fracaso de cada maniobra de enlace [Valdés *et al.*, 2011c], [Valdés *et al.*, 2012a], [Valdés *et al.*, 2012b].
- Diseño de una plataforma de simulación y experimentación real para la ejecución de la estrategia de enrutamiento en un escenario de transporte con robots móviles reales Pioneer P3-DX [Valdés *et al.*, 2008], [Espinosa *et al.*, 2008a], [Espinosa *et al.*, 2008b], [Espinosa *et al.*, 2010], [Espinosa *et al.*, 2011], [Santos *et al.*, 2010].

#### 1.4. Estructura de la tesis

El trabajo ha sido dividido en 6 capítulos. En el capítulo 2 se analiza el estado del arte respecto a los algoritmos de enrutamiento más utilizados en el campo de la robótica móvil y en el ámbito del transporte terrestre en general. Se revisa la utilización de técnicas de Programación Dinámica a la resolución de problemas de cálculo de trayectorias.

El capítulo 3 aborda el tema principal de este trabajo, o sea, la estrategia de enrutamiento que garantiza que un vehículo perseguidor pueda incorporarse a un convoy en movimiento, cumpliendo determinadas restricciones temporales. Para ello se presenta una solución al cálculo del nodo de enlace entre seguidor y convoy, y de un algoritmo de enrutamiento eficiente para llegar hasta él, teniendo en cuenta que la metodología y algoritmos propuestos son trasladables a otros problemas de enrutamiento más generales.

En el capítulo 4 se modela la incertidumbre de los tiempos de recorrido de cada tramo y se analizan sus consecuencias en el cálculo del nodo óptimo y la ruta de enlace. Se presenta el término “factor de riesgo” y se evalúan sus efectos en la fiabilidad del nodo y ruta obtenidos. Además, se plantea el procedimiento de reevaluación dinámica del nodo de enlace y de la actualización de los tiempos de recorrido de los tramos del entorno cada vez que por ellos circula una unidad.

El capítulo 5 presentan los resultados mediante el demostrador software diseñado para validar las estrategias propuestas anteriormente, para ello se recurre a la herramienta de simulación Player/Stage. Además, la algoritmia se implementa en un demostrador real, a partir de un mapa artificial diseñado al efecto y utilizando como prototipo de unidad de transporte unidades robóticas Pioneer P3-DX.

Por último, en el capítulo 6 se muestran las conclusiones de la tesis y se proponen líneas de trabajo futuro que permitan mejorar y/o complementar los resultados obtenidos en la misma.

# 2

## ESTADO DEL CONOCIMIENTO

---

Este capítulo muestra el estado del arte de las estrategias de enrutamiento más extendidas en el campo de la navegación en la robótica móvil. Se mencionan los algoritmos con mayor presencia en la literatura actual, caracterizándolos, analizando sus principales aspectos y mostrando su evolución hacia métodos más completos y eficientes. También se comentan algunas de las herramientas actuales que más se emplean en la búsqueda de rutas.

### 2.1. Revisión de algoritmos de enrutamiento

A continuación se revisa el estado del arte de las principales estrategias de enrutamiento. El problema de encontrar un camino efectivo entre dos o más puntos en un mapa ha sido un tema recurrente en las últimas décadas, siendo uno de los principales motivos de investigación en el campo de los Sistemas de Transporte Inteligentes [Sanders *et al.*, 2007], [Song *et al.*, 2010]. A día de hoy, herramientas comerciales como los navegadores TomTom, o aplicaciones web como Google Maps, NAVTEQ y otras permiten encontrar la mejor ruta entre origen y destino atendiendo a diferentes criterios de elección, como por ejemplo la distancia total de la ruta o el tiempo de viaje a la velocidad máxima permitida.

En este capítulo se realiza una revisión bibliográfica acerca de los principales algoritmos de enrutamiento reflejados en la literatura científica actual, los que sin duda constituyen la base teórica para muchas de estas aplicaciones. En primer lugar se plasmarán algunas definiciones necesarias para una mejor caracterización y comprensión de los algoritmos de enrutamiento que se describen en este capítulo. Para una mayor claridad en este estudio, los métodos analizados se han clasificado en tres categorías principales, atendiendo a su nivel de complejidad y de la información que utilizan en la búsqueda: estáticos, dinámicos, y de tiempo real.

En el contexto de la navegación, se puede definir el enrutamiento como el proceso de búsqueda para obtener caminos válidos en un grafo, desde un nodo inicial a uno final. Un grafo (del griego *grafos*: gráfica o dibujo) es informalmente hablando, una representación simplificada de una red, como puede ser por ejemplo el mapa de una ciudad u otra zona geográfica de interés. Desde un punto de vista matemático, un grafo  $G(V, E)$  está formado por un conjunto  $V$  de nodos, generalmente finito, y un conjunto de arcos  $E$  que conectan a estos nodos entre sí, denominándose un par de nodos *adyacentes* a aquellos que estén

conectados por un arco común.  $G$  es un grafo *dirigido* si sus arcos muestran algún tipo de direccionalidad, que generalmente es representada mediante flechas y que indican la dirección del camino, en caso contrario, se dice que es *no dirigido*, y a efectos prácticos se considera bidireccional. Puede deducirse que un camino en un grafo que representa una ciudad no es más que una sucesión de nodos (intersecciones) y arcos adyacentes (calles), dispuestos según la dirección de la ruta.

La vinculación entre nodos de un grafo es mediante la relación padre-hijos, o antecesor-sucesores, donde los segundos se obtienen a partir de los primeros mediante un proceso conocido como *expansión*, que permite que una parte del grafo sea almacenada en memoria para su análisis. Así, se dice que un nodo padre se *expande* cuando se *generan* sus descendientes. Este proceso de expansión intuitivamente crea una representación del grafo comúnmente conocida como árbol de búsqueda que, como puede observarse en la Figura 2.1, caracteriza la parte explorada del grafo en un momento dado. En un árbol cada par de nodos adyacentes tiene solo un camino entre ellos, pudiendo accederse a un nodo desde únicamente otro nodo. Dicho de otro modo, cada nodo tendrá solamente un padre, excepto el nodo origen o raíz que no tendrá ninguno. Este tipo de representación en forma de árbol hace imposible la existencia de ciclos de búsqueda infinitos [Luger, 2005] y de esta forma se evita perder tiempo expandiendo nodos que ya han sido expandido antes [Russell *et al.*, 2003].

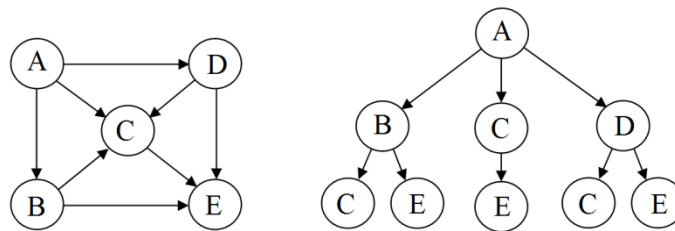


Figura 2.1. Grafo dirigido (izda) y su árbol correspondiente (dcha), comenzando la expansión en el nodo A.

Para almacenar un grafo en memoria se suelen emplear dos estructuras estándar: mediante un arreglo de *listas de adyacencia* [Leu *et al.*, 2004], [Ma *et al.*, 2009] o mediante *matrices de adyacencia* [Bang-Jensen *et al.*, 2007]. La primera constituye una forma de representación más compacta y se emplea para grafos más bien con un bajo nivel de ramificación, por ejemplo aquellos en los que se cumple que:  $E \ll V^2$  [Cormen *et al.*, 2009]. En este caso cada nodo del grafo tiene asociada una lista que contiene todos los nodos adyacentes al mismo, de modo que para un grafo de  $N$  nodos, se tendrán también  $N$  listas. Por su parte, las matrices de adyacencias constituyen la representación preferida para grafos más densos [Ghoniem *et al.*, 2005] en los que el conjunto  $E$  es más cercano a  $V^2$ . Así, el grafo completo se representará con una matriz de  $N \times N$  elementos, donde  $n_{i,j}$  será igual a 1 en caso de existir un arco que una al nodo  $i$  con el nodo  $j$ , y cero en caso contrario, como muestra la Figura 2.2. Se puede deducir que una de las principales desventajas de usar matrices de adyacencias radica en su alto consumo de memoria cuando se emplean para representar los nodos de grafos muy grandes, independientemente del número de arcos en el grafo.

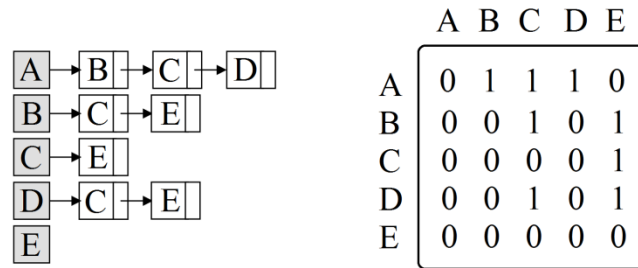


Figura 2.2. Grafo de la Figura 2.1 representado con listas y matriz de adyacencia, respectivamente.

Uno de los paradigmas fundamentales dentro del campo de los algoritmos de búsqueda de rutas es el llamado Problema del Camino más Corto (*The Shortest Path Problem, SPP*) [Wagner *et al.*, 2007], en realidad constituido por un conjunto de problemas de optimización, estudiados desde 1960 y cuyos métodos de resolución han sido constantemente mejorados, siendo uno de sus campos de aplicación más interesante el relacionado con el transporte.

Dentro de esta familia, y debido al amplio espectro de aplicaciones que poseen, los diferentes enfoques se han flexibilizado y han ganado en complejidad, pudiéndose mencionar dentro de los más importantes, el *SPP* aplicado a grafos estáticos, cuyos costes no varían con el tiempo [Delling *et al.*, 2009], el *SP-SPP* (*Single Pair Shortest Path Problem*) que se centra en el cálculo de rutas óptimas entre dos nodos específicos en un grafo [Sedgewick *et al.*, 1986], [Zhan *et al.*, 2000], el *SS-SPP* (*Single Source Shortest Path Problem*) para calcular rutas entre un nodo específico y los restantes nodos del grafo [Cantone *et al.*, 2009], el *TD-SSP* (*Time Depend Shortest Path Problem*) [Dreyfus, 1969], [Sung *et al.*, 2000], [Ding *et al.*, 2008], aplicado a grafos dinámicos, donde los costes de los arcos varían con el tiempo [Wagner *et al.*, 2005], [Osvald *et al.*, 2008], [Dell'Amico *et al.*, 2008] y otros como el *R-SPP* (*Restricted Shortest Path Problem*) que incorpora condicionantes de diverso índole a la búsqueda [Jocksch, 1966], [Azi *et al.*, 2007], [Ruzika *et al.*, 2011].

Para dar solución a estos problemas de enrutamiento aplicados a grafos, a día de hoy existe gran diversidad de algoritmos. Aunque existen varias formas de clasificación, en esta Tesis se clasifican atendiendo a su manera de operar, y se agruparán en tres categorías: algoritmos estáticos, dinámicos y de tiempo real.

Los algoritmos estáticos son aquellos que evalúan grafos donde los costes de sus arcos no varían a lo largo del tiempo y el cálculo de la ruta se realiza antes de dar el primer paso en la ejecución de la misma. Esto quiere decir que la ruta calculada será indiferente a las posibles variaciones en los costes del grafo que ocurran posteriormente.

Por su parte, en las estrategias dinámicas, el hecho de que los costes del grafo sean cambiantes obliga a realizar sucesivas búsquedas en la medida en que el mapa se va modificando, posibilitando la corrección de la ruta original cuando se detectan estos cambios que pueden consistir por ejemplo, en la presencia (o ausencia) de nuevos obstáculos que afecten la trayectoria a seguir y que no fueron contemplados en la búsqueda inicial. Finalmente y más evolucionados que los dos primeros grupos, en los algoritmos de

enrutamiento en tiempo real, se intercalan etapas de cálculo de la ruta con etapas de ejecución de la misma, por lo que a diferencia de los métodos anteriores, estas estrategias se aplican a casos en que el vehículo o robot móvil re-planifica su trayectoria a medida que la va recorriendo, actualizándola o corrigiéndola de acuerdo a la posible nueva información descubierta a cada instante, como puede ser una lectura de láser o ultrasonido indicando un nuevo obstáculo en la dirección del movimiento. El cálculo parcial de la ruta completa permite que el algoritmo de enrutamiento sea lo suficientemente rápido en hallar una solución y evite importantes demoras que afecten al recorrido del vehículo.

Otro aspecto importante a la hora de analizar algoritmos de enrutamiento es el concerniente a los criterios que miden su eficiencia, y que comúnmente son empleados para diferenciar y comparar unos métodos de otros, considerando que lo deseable en un algoritmo es que su desempeño no dependa del soporte computacional en el que se ejecute, sino de la propia naturaleza del algoritmo y de su desarrollo durante el proceso de búsqueda de la solución. En general, los criterios más extendidos para definir la eficiencia de un algoritmo respecto a otro son:

- **Complejidad temporal:** Mide el tiempo requerido por un algoritmo en desarrollar la búsqueda hasta encontrar una solución. El coste temporal es directamente proporcional al factor de ramificación del árbol de búsqueda y de la profundidad a la que se encuentre la solución. Cuanto más complejo sea el árbol, más tiempo empleará el algoritmo.
- **Complejidad espacial:** Durante el proceso de búsqueda el algoritmo almacena información de parte o la totalidad de los nodos del árbol, variando grandemente entre distintos métodos de búsqueda, en dependencia de si esta es más o menos exhaustiva. La cantidad de memoria necesaria para almacenar esta información dependerá del tipo de estructura de datos que emplee cada método, y de la manera en que se accede a los mismos. Aunque con el desarrollo creciente de ordenadores más potentes esta restricción es cada día menos tomada en cuenta, aún es un factor importante.
- **Complejidad:** Un algoritmo es completo si garantiza que encontrará al menos una solución al problema, si esta existiese.
- **Optimalidad:** Un algoritmo será considerado óptimo si garantiza que encontrará la mejor solución de entre todas las posibles soluciones del problema. Aunque tal vez sea la característica más deseable en un algoritmo de búsqueda, en muchos casos está intrínsecamente relacionada con el coste temporal y espacial del algoritmo, siendo inversamente proporcional a estos, por lo que en muchas ocasiones se renuncia a esta en pos de la eficiencia computacional del método.

### 2.1.1. Estáticos y dinámicos

Una primera definición de los algoritmos de enrutamiento se puede dar de acuerdo al comportamiento temporal de la información del mapa disponible. Así, podemos definir dos grupos de métodos de enrutamiento, los estáticos en los que, como se mencionó antes, los

costes de los arcos del grafo analizado no varían a lo largo del tiempo y el cálculo de la ruta se realiza antes de dar el primer paso en la ejecución de la misma, haciéndola indiferente a las posibles variaciones en los costes del grafo que ocurran posteriormente, y los dinámicos, que actualizan parte de la información durante el propio proceso de búsqueda. A continuación se estudiarán los métodos más representativos de cada grupo.

### **Algoritmos de enrutamiento estático:**

Por su sencillez, el algoritmo **Primero en Anchura** (*Breadth-First Search* o *BFS*) formulado por Moore en 1959 [Moore, 1959], es uno de los más fáciles de entender e implementar, y es uno de los métodos de búsqueda representativo de la familia de los no informados o “ciegos”, llamados así porque llevan a cabo el proceso de búsqueda siguiendo un orden totalmente independiente de la naturaleza de la solución buscada. En ellos, la ubicación y características del nodo destino dentro del espacio de búsqueda no altera el orden de expansión de los nodos, que está basado en una secuencia de pasos rígidos realizados cíclicamente, hasta que la solución es encontrada, si esta existe.

El proceso de búsqueda comienza con un árbol que inicialmente contiene solo el nodo raíz, el cual es expandido para generar sus sucesores, luego los sucesores de estos y así continuamente, como se muestra en la Figura 2.3, donde los números indican el orden de expansión de los nodos. Como puede observarse, la búsqueda se lleva a cabo por niveles, analizando todos los nodos de un nivel antes de pasar al siguiente, y continuando hasta que el nodo final es encontrado. Para almacenar en memoria los nodos que van siendo generados a cada instante, el algoritmo emplea una estructura de datos de tipo *FIFO* (*First In-First Out*), llamada cola de prioridad o lista *ABIERTA*, donde los primeros nodos que se generen serán también los primeros que se expandan, y así capa por capa hasta encontrar el nodo objetivo.

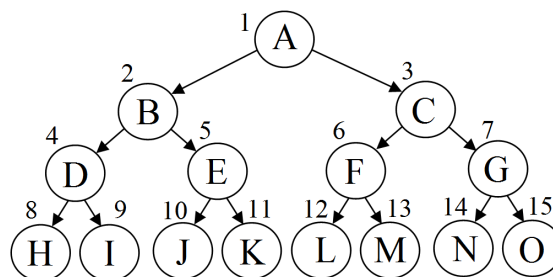


Figura 2.3. Algoritmo Primero en Anchura (BFS). Los nodos son enumerados según el orden de análisis.

Debido a que realiza una búsqueda exhaustiva, el algoritmo se considera completo pues siempre encontrará una solución si existe, y, siempre que todos los costes sean iguales, tendrá carácter óptimo [Russell *et al.*, 2003], garantizando que una vez encontrado el nodo final se ha hecho a través del camino más corto. Con todo, la principal desventaja de este algoritmo se puede deducir fácilmente si nos fijamos en el orden de expansión de los nodos, pues hasta que la solución sea encontrada, todos los nodos expandidos deberán permanecer almacenados en memoria ya que se necesitan para reconstruir la ruta. De esta manera, en árboles muy grandes se tendrá un considerable consumo de recursos de memoria, de forma



que la complejidad espacial es del orden de  $O(b^{d+1})$ , según la notación planteada en [Cormen *et al.*, 2009], siendo  $b$  el número de sucesores por nodo y  $d$  el de niveles de profundidad en el árbol de búsqueda. A la misma expresión responde la complejidad temporal, por lo que su aplicación será muy poco eficiente en análisis a gran escala [Zhou *et al.*, 2006].

A diferencia del algoritmo anterior que realiza el análisis a todo lo ancho del grafo, el algoritmo **Primero en Profundidad** (*Depth-First Search o DFS*) [Pearl, 1984], [Russell *et al.*, 2003], realiza la búsqueda rama por rama, expandiendo el nodo no expandido más profundo de la rama que está siendo analizada, visitando tanto este como sus descendientes a la máxima profundidad posible, y retornando a la última posibilidad disponible si no ha encontrado ninguna solución. El procedimiento se ilustra en la Figura 2.4, mostrando el orden de expansión de los nodos.

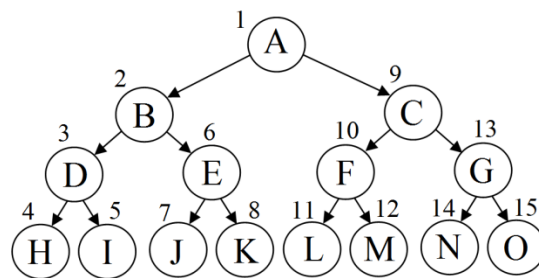


Figura 2.4. Algoritmo Primero en Profundidad (DFS).

La principal ventaja de la búsqueda en profundidad radica en que la exploración necesita muy poca memoria ya que no es necesario almacenar más nodos que aquellos de la rama que está siendo analizada en cada momento. Si todos los nodos de la rama explorada son expandidos sin que se encuentre la solución, se desecha la rama completa, eliminando los nodos de la memoria, y se analiza la rama contigua. Debido a su forma de operar, la estructura empleada para almacenar los nodos, será de tipo *LIFO* (*Last In-First Out*), es decir, una pila, donde el último que se genera es el primero que se expande. A diferencia de la búsqueda en anchura, en este algoritmo la complejidad temporal será de  $O(b^d)$  en el peor de los casos, y de  $O(d)$  la espacial [Joshi *et al.*, 2006]. Sin embargo, el realizar una búsqueda en profundidad, rama por rama, tiene dos inconvenientes importantes. El primero de ellos es que la búsqueda puede verse envuelta en ciclos infinitos cuando se analizan ramas muy profundas que tal vez no llevan a la solución, la que por otra parte puede encontrarse a mucha menos profundidad, en una rama contigua no analizada aún. Esto significa que se corre el riesgo de consumir tiempo de más en búsquedas estériles. El segundo problema radica en que por esta misma causa, aunque el algoritmo es completo, no asegura que la solución encontrada se haga a través del camino más corto, por lo que no es un algoritmo óptimo.

Para resolver el problema de caer en ciclos muy largos o infinitos de los que puede ser difícil e incluso imposible salir cuando se trabaja con árboles muy profundos se han conseguido algunas variantes que restringen la profundidad de la búsqueda. Una de ellas es el algoritmo **Primero en Profundidad con profundidad limitada** [Jing *et al.*, 2009], que realiza la

búsqueda hasta una profundidad específica  $l$ , excluyendo los nodos que se encuentran a mayor profundidad y por tanto reduciendo la cantidad de nodos a analizar. De esta forma, cuando la solución se encuentre a profundidad menor o igual a  $l$ , se empleará menos tiempo en encontrarla, siendo su complejidad temporal y espacial del orden de  $O(b^l)$  y  $O(l)$ , respectivamente. Con todo, si la solución se encuentra fuera de los límites de profundidad fijados, el algoritmo no devolverá nada. El algoritmo **Primero en Profundidad con profundidad iterativa** [Korf, 1985], intenta resolver este problema ejecutando la búsqueda una y otra vez mientras aumenta progresivamente este límite de profundidad en cada nueva iteración, cuando no encuentra la solución a la profundidad actual. Este algoritmo combina las ventajas de la búsqueda en anchura, siendo completo cuando el grafo es finito, y la búsqueda en profundidad por los pocos recursos de memoria que consume. Su complejidad temporal es del orden  $O(b^d)$ , y de  $O(d)$  la espacial, donde a diferencia de la variante básica,  $d$  aumenta progresivamente hasta encontrar la solución.

Los métodos analizados hasta ahora realizan la búsqueda partiendo únicamente del nodo origen en una búsqueda en una única dirección. Sin embargo, el algoritmo de **búsqueda bi-direccional** [Pohl, 1969], lleva a cabo dos búsquedas simultáneas, una descendente a partir del nodo inicial y otra que asciende desde el nodo objetivo (véase Figura 2.5), para encontrar la solución en un tiempo significativamente menor, considerando que es más eficiente hacer dos búsquedas pequeñas de complejidad  $O(b^{d/2})$  que una única más grande de complejidad  $O(b^d)$ . El proceso finaliza cuando ambas búsquedas se encuentran en un nodo común, construyéndose la ruta a partir de la suma de ambos recorridos, garantizando la optimalidad y completitud del algoritmo si ambas búsquedas son hechas Primero en Anchura [Russell *et al.*, 2003].

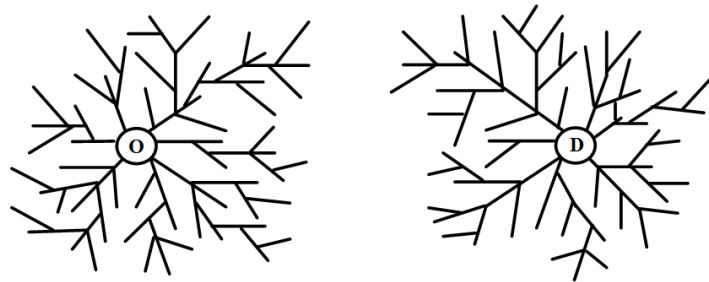


Figura 2.5. Algoritmo de búsqueda bi-direccional, a partir del origen (izda) y a partir del destino (dcha).

Además, como condición para su implementación se debe tener conocimiento explícito del nodo solución para poder comenzar la segunda búsqueda en él, y por supuesto evitar la repetición de nodos contando con un método eficiente que identifique, en cada proceso, si un nodo cualquiera a analizar ya está formando parte de otra búsqueda.

Los algoritmos vistos hasta el momento siguen un esquema simple y secuencial, y pueden dar resultados aceptables cuando se aplican a grafos poco complejos. Sin embargo, al no tener en cuenta información relevante que ayude a orientar la búsqueda en la dirección correcta, algunos de ellos pueden llegar a realizar una exploración exhaustiva del grafo, llegando incluso a recorrerlo en su totalidad, por lo que al aumentar el tamaño del problema,

presentan importantes carencias en su desempeño, siendo una de las principales el crecimiento exponencial en el consumo de recursos de memoria y tiempo necesarios para realizar la búsqueda, por lo cual no son aconsejables particularmente en grafos extensos, con miles de nodos, como perfectamente puede implicar el representar una ciudad genérica.

En este sentido y con el objetivo de solventar estas dificultades, los algoritmos descritos a continuación, utilizan información adicional del grafo y del nodo destino para orientar la búsqueda en la dirección de la solución. En general, en estos algoritmos los nodos están caracterizados por una función de evaluación  $f$  que indica cuan bueno puede llegar a ser un nodo para formar parte de la ruta óptima. Así, es posible discriminar muchos nodos que no intervienen en la solución y por tanto que su análisis no es de interés, logrando reducir el coste espacial o temporal que cada algoritmo invierte en el proceso de búsqueda. La gran mayoría de algoritmos de búsqueda y sus variantes se encuentran en este grupo y es a los que se les dedicará una mayor atención en adelante, organizando su análisis de acuerdo a la complejidad de los mismos.

Dentro de esta clasificación se puede mencionar al Algoritmo **Dijkstra** [Dijkstra, 1959], que constituye uno de los más populares algoritmos de búsqueda y planificación de rutas [Kong *et al.*, 2011], [Wang *et al.*, 2011] y tal vez el más completo desde el punto de vista teórico. Este algoritmo calcula el camino más corto entre un par de nodos origen-destino en un grafo, explorando todos los posibles caminos entre el origen y cada uno de los restantes nodos del grafo, expandiendo siempre el nodo más cercano al origen y parando cuando encuentra la solución.

Para ello, cada nodo  $n$  del grafo tiene asociada una función de evaluación  $f(n) = g(n)$ , donde  $g(n)$  es el coste del camino recorrido desde el origen al nodo en cuestión. Comenzando en el nodo inicial, el procedimiento genera los sucesores de este y los almacena en una lista *ABIERTA*, de nodos por analizar, ordenados de forma ascendente según el coste  $f(n)$ . El siguiente nodo a expandir será el primero en la lista, generando sus sucesores y repitiéndose el proceso hasta que se alcanza el nodo final. La búsqueda es completa y óptima siempre y cuando el coste de la ruta nunca disminuya conforme se avanza por ella, lo que equivale a decir que el árbol solo puede tener costes positivos.

De esta manera, se asegura siempre encontrar el nodo destino a través del camino mínimo o de menor coste, haciéndolo un algoritmo óptimo. Sin embargo, Dijkstra inicialmente almacena en la lista *ABIERTA* todos los nodos del grafo, empleando una matriz de adyacencia de dimensión  $N \times N$ , siendo  $N$  el número total de nodos, de modo que si el grafo es extenso la búsqueda puede volverse ineficiente y consumir mucha memoria y tiempo hasta encontrar la solución [Chao, 2010]. Además, debido a que la forma de operar del algoritmo es expandiendo nodos concéntricamente desde el origen, una gran parte de los que se exploran ni siquiera se encontrarán en la dirección del nodo de destino y no formarán parte de la solución final.

A pesar de estas desventajas, el algoritmo Dijkstra es uno de los más utilizados, y cuenta en la actualidad con diversas variantes en su empleo centradas en mejorar su eficiencia. Algunos

autores proponen para reducir el consumo de memoria en la búsqueda limitar el espacio total a una zona específica del mapa, rectangular [Fan *et al.*, 2010] o hexagonal [Nazari *et al.*, 2008], restringiendo el número de nodos a visitar por el algoritmo. Aunque se logra reducir considerablemente el consumo de recursos, el hecho de limitar el área total original tiene el inconveniente de que en ocasiones la trayectoria óptima no pueda ser obtenida, por quedar esta fuera del área de búsqueda.

[Noto *et al.*, 2000] aplica el concepto de búsqueda bi-direccional al Dijkstra, comenzando simultáneamente tanto desde el nodo inicial como desde el nodo destino, y restringiendo además el espacio de búsqueda. El resultado logrado también reduce significativamente el tiempo de búsqueda, permitiendo su aplicación en problemas de enrutamiento que requieren soluciones en tiempo real. Otros trabajos [Fan *et al.*, 2010], [Chao, 2010], [Gao, 2010], demuestran que es posible aumentar la eficiencia del Dijkstra optimizando la forma de almacenar la información, proponiendo como estructuras de datos el empleo de listas de adyacencia para guardar la información topológica del mapa de un modo más simple que mediante el método habitual con matrices de adyacencia. Esto, conjuntamente con un pre-tratamiento de la información o restringiendo dinámicamente el área de búsqueda, logra reducir los recursos de memoria y el tiempo de procesamiento del algoritmo. Maue particiona el grafo en conjuntos y pre-calcula la distancia entre los nodos de frontera de cada uno de ellos para acelerar la búsqueda [Maue *et al.*, 2006], mientras [Ma *et al.*, 2009] reduce el tiempo de ejecución significativamente empleando una estructura de datos de tipo K-ary (*K-ary heap*) y una lista de adyacencia inversa para reducir la complejidad de la búsqueda. Combinado con otros métodos [Kang *et al.*, 2008] como la optimización de enjambre de partículas (*Particle Swarm Optimization*), se logran igualmente resultados óptimos. Con todo, el Dijkstra solo es óptimo y completo en grafos con pesos no negativos. Para el caso contrario, Bellman y Ford desarrollaron el algoritmo que lleva sus nombres [Bellman, 1958], [Ford, 1956], [Golden, 1976], un método que calcula el camino más corto donde los pesos de los arcos pueden ser negativos, siendo una generalización del algoritmo de Dijkstra.

La búsqueda **Primero el Mejor** (*Best-First Search*) es uno de los primeros métodos que utilizan información heurística para mejorar la eficiencia de la búsqueda. En este algoritmo, se trata de expandir siempre el nodo más cercano al objetivo, para lo cual la función de coste asociada a cada nodo del grafo es  $f(n) = h(n)$ , donde  $h(n)$  indica el coste estimado de alcanzar el nodo final desde el nodo actual; a diferencia del Dijkstra, que emplea el coste desde el origen a cada nodo. En el proceso de búsqueda se emplean dos estructuras de datos, una lista *ABIERTA* en las que se guardan los nodos que han sido generados pero no analizados y una lista *CERRADA* en las que se guardan los nodos que ya han sido analizados. La búsqueda parte del nodo inicial, generando sus hijos y almacenándolos en la lista *ABIERTA*. A cada nodo  $n$  almacenado en *ABIERTA* se le asigna la función  $h(n)$  con el coste estimado de alcanzar el nodo final desde el mismo. Seguidamente el nodo inicial se guarda en *CERRADA*. El procedimiento continúa extrayendo para su análisis, el nodo de la lista *ABIERTA* con menor coste, generando sus sucesores y repitiendo el proceso hasta que el nodo destino es encontrado.

Sin embargo, este algoritmo no considera el coste del camino recorrido desde el inicio, sino solamente la estimación del camino que queda por recorrer, por lo que no es un algoritmo ni óptimo ni completo [Russell *et al.*, 2003]. Por otro lado, los requerimientos de memoria de este método son semejantes a los del algoritmo Primero en Anchura, por lo que la clave estará en elegir una buena función heurística que permita dar con una solución antes de que el algoritmo exceda los límites de memoria impuestos al proceso de búsqueda [Joshi *et al.*, 2006].

Presentado por primera vez en 1968 por Hart [Hart *et al.*, 1968] y popularizado posteriormente por Nilsson [Nilsson, 1971], el algoritmo  $A^*$  es seguramente el método de búsqueda heurística de uso más extendido [Goldberg *et al.*, 2004], [Xi *et al.*, 2006]. Este es una versión ampliada de la búsqueda Primero el Mejor, añadiendo al término heurístico el coste  $g(n)$  que ha implicado alcanzar cada nodo  $n$  desde el origen, tal como se aplica en Dijkstra, haciéndolo más completo que la búsqueda Primero el Mejor. Así, cada nodo es valorado por una función de coste de la forma  $f(n) = g(n) + h(n)$ . Por otro lado, siempre que el término heurístico cumpla ciertas características, el  $A^*$  se comportará de forma óptima, al igual que el Dijkstra, siendo la principal ventaja con respecto a este último que para llegar a la misma solución se requieran expandir muchos menos nodos, gracias a que la función heurística se emplea para dirigir el proceso de búsqueda en el sentido de la solución.

La optimalidad dependerá de la heurística empleada, que debe ser *admisibile* [Pearl, 1984], esto es, se ha de cumplir que en cada nodo analizado la función  $h(n)$  nunca sobre-estime el coste real de alcanzar el destino. Una forma muy común de establecer una heurística admisible es considerar como coste estimado la distancia euclídea entre el nodo en cuestión y el destino, pues evidentemente no hay nada más directo que la recta que los une por lo que se asegura que  $h(n)$  no sobre-estime el coste real de alcanzar el nodo final. De todas formas, la admisibilidad asegura la optimalidad del algoritmo pero no su eficiencia computacional, que dependerá en gran medida de la calidad de la heurística empleada [Cazenave, 2006]. Estos y otros aspectos serán tratados más adelante en el capítulo 3 de esta Tesis.

Considerando que, al igual que el Dijkstra, el  $A^*$  es un algoritmo óptimo cuando emplea heurísticas admisibles, la mayoría de trabajos encontrados en la literatura centran sus esfuerzos en mejorar la eficiencia del algoritmo, ya sea reduciendo tanto el tiempo de ejecución como la memoria empleada en el proceso de búsqueda [Liu *et al.*, 2011], [Lin *et al.*, 2008]. Incluso algunos autores, bajo la premisa de que en la gran mayoría de las veces basta con obtener una solución sub-óptima (aunque cercana a la óptima), han desarrollado versiones del  $A^*$  que sacrifican optimalidad en pos de un mejor rendimiento computacional como el **Weighted  $A^*$**  [Pohl, 1970], [Pohl, 1973], el  $A_\epsilon^*$  [Pearl *et al.*, 1982], o la **búsqueda optimista** de [Thayer *et al.*, 2008], [Thayer *et al.*, 2010].

El algoritmo **IDA\*** (*Iterative-Deepening  $A^*$* ) propuesto por [Korf, 1985] es una versión del  $A^*$  que emplea una estrategia similar a la búsqueda con profundidad iterativa, siendo óptimo y completo como el primero pero donde la complejidad espacial pasa de ser exponencial ( $A^*$ ) a lineal (IDA\*) [Korf, 1987]. La búsqueda se realiza comenzando en el nodo inicial y

continuando hasta una profundidad determinada por un umbral construido en base a información heurística. Si la búsqueda excede el valor del umbral sin encontrar la solución, la rama analizada es desechada y se pasa a la siguiente. Si al analizar todas las ramas no se ha dado con la solución, se re-expande la búsqueda hasta una nueva profundidad, incrementado el umbral según el coste de los nodos de frontera del árbol. La dificultad de este método radica en que, aunque se ahorra en recursos de memoria, re-expandir la búsqueda repetidas veces consecuentemente incrementa el tiempo de ejecución del algoritmo. La solución al problema de muchas re-expansiones puede ser incrementar el umbral desde un inicio a unos límites razonables en los que se complete la búsqueda en un tiempo razonable.

### **Algoritmos de enrutamiento dinámico**

El algoritmo **LPA\*** (*Lifelong Planning A\**) [Koenig *et al.*, 2002a] es una versión incremental del algoritmo **A\***, que combina la búsqueda incremental (Algoritmo *Dynamic SWSF-FP* [Ramalingam *et al.*, 1996]) y la búsqueda heurística (**A\***). Su principal ventaja radica en que, una vez obtenida la ruta, si ocurriesen cambios en los costes de los arcos del grafo, es capaz de calcular nuevamente la ruta reutilizando las partes del árbol original que no han sufrido variaciones, sin necesidad de repetir el proceso desde el comienzo. Por este motivo, la primera búsqueda es idéntica a la realizada con el **A\*** pero las siguientes son mucho más rápidas.

A pesar de esta ventaja, el **LPA\*** tiene el problema de que el número de nodos a expandir en cada búsqueda puede variar grandemente, en dependencia de la ubicación en el grafo de los nodos que han cambiado, llegando en ocasiones a ser excesivamente alto. En este sentido existen propuestas para mejorar el rendimiento ya sea mejorando la heurística y reduciendo el espacio de búsqueda [Huang *et al.*, 2007], u obteniendo la información del mapa a distintas escalas lo que reduce considerablemente la complejidad computacional de la búsqueda y del número de nodos expandidos en esta [Lu *et al.*, 2011].

Uno de los algoritmos de enrutamiento más difundido es la versión dinámica del **A\***, **Dynamic A\*** o **D\*** [Stentz, 1994], [Stentz, 1996], [Halaoui, 2009], orientado al enrutamiento en entornos cambiantes que son parcial o totalmente desconocidos. Un ejemplo de situaciones de este tipo puede ser aquella en las que un robot móvil debe moverse por un entorno inexplorado apoyándose en sus sensores para detectar posibles obstáculos en su camino. En este tipo de tarea, el algoritmo es capaz de obtenerla mejor ruta hasta el objetivo, utilizando la información disponible en el momento del cálculo y re-planificando si al recorrerla se detectan cambios en el entorno como la presencia o ausencia de obstáculos, añadiendo esta nueva información a la nueva búsqueda. De esta manera, la ruta es recalculada desde el estado actual del robot por lo que no hay que generar la ruta completa desde el inicio.

La versión básica del **D\*** tiene el inconveniente de que, al propagar los cambios ocurridos en los nodos del mapa, no se tiene en cuenta que alguno de estos constituyen estados inválidos por los que el robot no puede circular, y por tanto no deberían considerarse. Este problema es tratado por Stentz en una versión mejorada, el algoritmo **Focussed D\*** [Stentz, 1995],

incorporando una función heurística para enfocar el proceso de propagación de los nodos afectados y poder minimizar la expansión de los mismos, reduciendo también los costes computacionales de la operación. Por su parte, Koenig aprovecha las ventajas de la búsqueda incremental del algoritmo LPA\* para desarrollar **D\*Lite** [Koenig *et al.*, 2002b], [Koenig *et al.*, 2002c], una alternativa al Focussed D\* que implementa su misma estrategia de navegación pero el algoritmo es más simple de entender e implementar, y tan eficiente, al menos, como éste último.

### 2.1.2. Algoritmos de enrutamiento en tiempo real

Los algoritmos heurísticos tradicionales como el A\* obtienen la ruta óptima antes de que se produzca el primer movimiento del vehículo o robot móvil, y por ello son conocidos como métodos de búsqueda off-line. No obstante, algunos autores aplican estos métodos al problema del enrutamiento en tiempo real, donde en general se re-calcula la ruta cada vez que un nuevo nodo de la misma es alcanzado por el vehículo, incorporando información actualizada todo el tiempo [Saizheng *et al.*, 2009]. Esta información puede ser de varios tipos, pudiendo constituir tiempos de recorrido, número de vehículos circulando por un tramo determinado, etc [Khosroshahi *et al.*, 2011].

Sin embargo, el hecho de calcular la ruta completa repetidas veces implica que en mapas muy extensos se puede utilizar un tiempo innecesario en cada búsqueda. Es por ello que la mayoría de métodos de búsqueda en tiempo real, en vez de calcular la ruta completa, se centran en obtener solamente los pasos inmediatos que debe tomar el vehículo para alcanzar su objetivo [Bulitko *et al.*, 2008], llevando a cabo una sucesión de cálculos parciales y ejecuciones de los mismos por parte del robot, hasta alcanzar el nodo final.

Uno de los principales exponentes de esta última categoría es el algoritmo **RTA\*** (*Real-Time A\**), propuesto por [Korf, 1990]. Este algoritmo basa su ejecución en dos etapas diferentes e intercaladas. En la primera fase, de planificación, se realiza una búsqueda con profundidad limitada, empleando un término heurístico para guiar la búsqueda y elegir los siguientes pasos a tomar en el entorno cercano a la ubicación actual. Las nuevas acciones para alcanzarlos se llevan a cabo entonces en la fase de ejecución, repitiéndose el ciclo hasta que el robot alcanza el objetivo. A diferencia del A\*, en el RTA\* el coste  $g(n)$  representa la distancia al nodo analizado  $n$  desde la posición actual del robot y no desde el nodo origen. El algoritmo es completo siempre que el espacio de búsqueda sea finito, tenga costes positivos, valores heurísticos finitos y el nodo final sea alcanzable desde cualquier otro nodo [Korf, 1990]. Sin embargo, no asegura la optimalidad de las soluciones, aunque sí que las soluciones sub-óptimas pueden ser alcanzadas en mucho menos tiempo que con los tradicionales algoritmos de búsqueda off-line. Esto se debe a que, al estar limitada la profundidad de la búsqueda, se asegura que el tiempo requerido en cada una de ellas sea constante.

El mismo autor ha implementado una variante de este algoritmo que se comporta de manera similar pero que incorpora aprendizaje al proceso de búsqueda. El **LRTA\*** (*Learning Real-*

*Time A\**) es uno de los algoritmos de enrutamiento en tiempo real más conocidos [Korf, 1990], [Bulitko *et al.*, 2006]. Este, al igual que RTA\* intercala etapas de planificación y ejecución, pero tiene la propiedad de actualizar el término heurístico de cada nodo del espacio de búsqueda al realizar repetidas exploraciones, haciendo la heurística más y más exacta hasta que eventualmente converge al coste real hasta el objetivo. El LRTA\* es completo bajo las mismas condiciones del RTA\* descritas anteriormente, y asegura la optimalidad de la solución en un número finito de pruebas, siempre que la heurística inicial sea admisible [Bulitko *et al.*, 2006].

Bond [Bond *et al.*, 2010] presenta un algoritmo de búsqueda bidireccional que combina las ventajas de la búsqueda dinámica y la de tiempo real. Su algoritmo RTD\* (*Real Time D\**) lleva a cabo una búsqueda dinámica estándar como la del D\*Lite, pero interrumpida periódicamente cada vez que el vehículo o robot móvil se mueve como sucede en la búsqueda en tiempo real del LRTA\*, mostrando muy buenos resultados en entornos dinámicos.

Otros métodos de búsqueda en tiempo real menos conocidos que el LRTA\* son el  $\epsilon$ -LRTA\* y el FALCONS. El  $\epsilon$ -LRTA\* [Ishida *et al.*, 1996], [Shimbo *et al.*, 2003] reduce el tiempo de búsqueda, obteniendo soluciones sub-óptimas pero que convergen más rápidamente y requieren menos memoria en la ejecución del algoritmo. La sub-optimalidad de las soluciones se consigue escalando la admisibilidad de la heurística inicial utilizada mediante un factor  $(1 + \epsilon)$ , siendo  $\epsilon$  el error que acota la optimalidad de la solución obtenida. Por lo que variando la admisibilidad de la heurística se varía la optimalidad de la solución, resultando rutas que se acercan en mayor o menor medida a la mejor en dependencia del grado de exigencia del problema tratado. Si  $\epsilon = 0$  entonces el algoritmo se comportará de forma idéntica al LRTA\*.

Por su parte, y a diferencia del método anterior que obtiene soluciones sub-óptimas, el algoritmo FALCONS (*FAst Learning and CONverging Search*) [Furcy *et al.*, 2000], [Hernández *et al.*, 2005], es un método que sin sacrificar la optimalidad de la solución, logra que la misma llegue a converger en un tiempo significativamente menor que el logrado por el LRTA\*, cambiando la manera en que se eligen los nodos en el proceso de búsqueda, logrando búsquedas que son hasta un 60 % más rápidas que las logradas por el LRTA\*.

### 2.1.3. Enrutamiento con incertidumbre en los tiempos de recorrido

Frecuentemente, en problemas de enrutamiento aplicados a casos reales, se deben definir los costes de los arcos en función del tiempo y no de la distancia de los mismos, ya que evidentemente esta última es fija y por tanto incapaz de reflejar cambios ocurridos en el mapa durante el recorrido de la ruta.

Los tiempos de recorrido de cada tramo van a depender de varios factores. Una forma simple de aproximarlos consiste en dividir la velocidad media del vehículo que recorrerá el tramo, entre la distancia de dicho tramo. Algunos autores hacen más compleja esta estimación



mediante otros factores como el tipo de vehículo, la hora del día, y día de la semana, entre otros, añadiéndole diferentes fuentes de incertidumbres a la información disponible.

De esta forma, a la hora de obtener rutas válidas es imprescindible tener en cuenta la variabilidad que sufren los tiempos de recorrido en los distintos tramos, tanto antes de calcular la ruta como durante el recorrido de la misma, siendo frecuentemente necesario realizar algún tipo de re-planificación para corregir la trayectoria calculada inicialmente.

La mayoría de trabajos publicados en la actualidad tienen en cuenta lo anterior y consideran las incertidumbres en los tiempos de recorrido, muchas veces modelándolos como una variable aleatoria que tiene una función de densidad de probabilidad con valor medio y varianza, que como se ha dicho antes, dependerá de factores como el día de la semana en que se realice el recorrido, la hora o el tipo de vehículo implicado [Noland *et al.*, 2002].

A modo de ejemplo, [Li, 2004] realiza un estudio en una sección de *CityLink*, una autopista de peaje en Melbourne, Australia, en el que analiza cómo varía la distribución de los tiempos de viaje en dependencia de diferentes ventanas de tiempo. Para ello, se registra el tráfico a lo largo de varios días y dentro de ellos, durante determinadas horas. El análisis estadístico demuestra que, cuanto menor es la ventana temporal, más la distribución de los tiempos de viaje se acerca a una distribución de probabilidad normal.

[Jula *et al.*, 2008] ha desarrollado métodos que permiten en una red de transporte estocástica y dinámica, predecir los tiempos de recorrido de los arcos de la misma. El predictor, basado en un filtro de Kalman, combina información obtenida en tiempo real con valores históricos de la red para obtener los tiempos de recorrido entre nodos, y además con estos valores estimar los tiempos de llegada a los distintos nodos de la red.

[Alazab *et al.*, 2011] propone un algoritmo de enrutamiento óptimo para redes de transporte estocásticas. El algoritmo constituye una variante del Dijkstra que considera parámetros variables en el tiempo, como la congestión del tráfico y la velocidad del vehículo que ejecuta la ruta, entre otros. Algunos de estos parámetros han sido obtenidos mediante un *GIS (Geographic Information System)* y otros son extraídos de una base de datos.

[Bartin *et al.*, 2010] lleva a cabo un estudio para identificar un conjunto de rutas óptimas en una red de autopistas de peaje en New Jersey, considerando información de viaje en tiempo real. La incertidumbre en los tiempos de recorrido se obtiene directamente registrando la varianza de los mismos durante un número determinado de días consecutivos.

El algoritmo empleado en esta tesis es una variante de algoritmo A\* clásico y utiliza como fuente de información heurística los tiempos estimados de recorrido entre nodos, sean estos consecutivos o no. Estos tiempos son actualizados en base a la experiencia de viaje de todas las unidades móviles que recorran el mapa en cualquier momento, estableciendo los valores estadísticos que caracterizan a cada uno de los tramos y zonas del mapa. Así, cuando una unidad móvil cualquiera recorre un tramo, se registra el tiempo empleado en hacerlo y se obtiene el valor medio y varianza del tiempo de recorrido del tramo. Ambos valores

estadísticos serán constantemente actualizados con los futuros recorridos de esta u otras unidades al pasar por el tramo en cuestión. Como se observará más adelante, la incorporación de esta información estadística va a tener una repercusión directa en la posibilidad de éxito o fracaso de la maniobra de enlace entre una unidad perseguidora y el convoy.

Por otra parte, a diferencia del A\* original que es ineficiente en grafos muy extensos, el algoritmo propuesto en esta tesis se caracteriza por ejecutar la búsqueda de la ruta de manera eficiente, computacionalmente hablando, empleando un mínimo de recursos de memoria y tiempo de búsqueda, como se mostrará en detalle en el siguiente capítulo.

## 2.2. Herramientas comerciales para el cálculo de rutas

Sin duda alguna la herramienta más conocida para el cálculo de rutas es el Google Maps, un servicio web gratuito lanzado en 2005 por la compañía Google que ofrece mapas e imágenes obtenidas por satélite, y permite calcular rutas entre un nodo origen y uno destino. Con el paso del tiempo, Google Maps ha ido mejorando las aplicaciones y aumentando la calidad y definición de los mapas que utiliza. Entre sus opciones más interesantes están la posibilidad de obtener rutas personalizables, por ejemplo que eviten autopistas o pasen por un punto específico de interés, el sugerir rutas alternativas a la óptima calculada, y más recientemente el mostrar el estado del tráfico en tiempo real, esto último resulta de gran interés en el acceso a las grandes urbes.

A diferencia de Google Maps que es un servicio web, existen dispositivos GPS que pueden ser instalados en cualquier vehículo como es el caso del TomTom Navigator. Esta tecnología, proporciona la ubicación exacta del vehículo y permite calcular la ruta más corta o rápida a un destino específico, manejando la información más actualizada disponible. Entre sus principales aportes se encuentra *HD Traffic*, un nuevo servicio que aglutina la información recibida desde millones de teléfonos móviles y dispositivos TomTom para conocer, en tiempo real, la velocidad y dirección de los flujos de tráfico en las redes de carreteras. Esto significa que cualquier usuario con un dispositivo TomTom o un teléfono móvil conectado al sistema puede aportar de forma anónima información en tiempo real sobre el estado de la vía por la que circula en ese instante. Toda esta información es utilizada en el cálculo de trayectorias óptimas siendo posible conocer con mucha más exactitud los tiempos de recorrido de las diferentes secciones de la ruta, posibles retrasos, vías cortadas o cualquier otra incidencia que pueda encontrarse en la misma y que pueda ser de utilidad para el enrutamiento.

Otras compañías que brindan servicios de localización y planificación de rutas son NAVTEQ y Garmin. Ambas compañías cuentan con sus propios mapas, software de localización y sus dispositivos de navegación GPS.

Sin embargo, estas herramientas están diseñadas para calcular trayectorias hasta destinos estáticos, a diferencia de nuestro caso en que se desea enlazar una unidad perseguidora a un convoy en movimiento y por tanto el nodo de destino es dinámico. Esto requiere una re-

evaluación constante del objetivo y, en caso necesario, la re-planificación de la ruta hasta el mismo. Para ello el algoritmo de enrutamiento empleado debe ser lo suficientemente eficiente para realizar el cálculo garantizando el enlace entre ambas unidades y considerando a su vez las incertidumbres en los tiempos recorrido de cada tramo del mapa.

## 3

## ENRUTAMIENTO ÓPTIMO Y EFICIENTE PARA MANIOBRAS DE ENLACE

Este capítulo describe la metodología seguida para llevar a cabo la maniobra de enlace entre una unidad perseguidora y un convoy en movimiento. Se detalla la solución propuesta para abordar un doble problema: cálculo del nodo óptimo de enlace y ruta eficiente para que la unidad perseguidora alcance con éxito dicho nodo.

### 3.1. Planteamiento del problema

El objetivo de esta tesis es proponer una estrategia de enrutamiento que garantice a una unidad móvil independiente, localizada en el interior de un entorno específico, unirse a un convoy que circula por la periferia del mismo, siguiendo una trayectoria dinámicamente actualizada. Para lograrlo, es necesario en primer lugar, encontrar el nodo más adecuado de la periferia que permita enlazar a la unidad independiente con el convoy, y en segundo lugar, calcular la mejor ruta hasta dicho nodo, todo ello cumpliendo determinadas condiciones de optimalidad y eficiencia. Aunque en este capítulo no se estudian las incertidumbres inherentes a los tiempos de recorrido tanto de la unidad independiente como del convoy cuando se mueven por el entorno, la estrategia global si las tiene en cuenta y serán estudiadas en el capítulo siguiente.

La primera parte de este capítulo describe el problema general a resolver, se establecen las condiciones iniciales del mismo, y se acotan y detallan todos los aspectos necesarios para la adecuada ejecución de la estrategia de enrutamiento propuesta. En la Figura 3.1 se resumen los elementos clave del sistema desarrollado, analizados con profundidad a continuación.

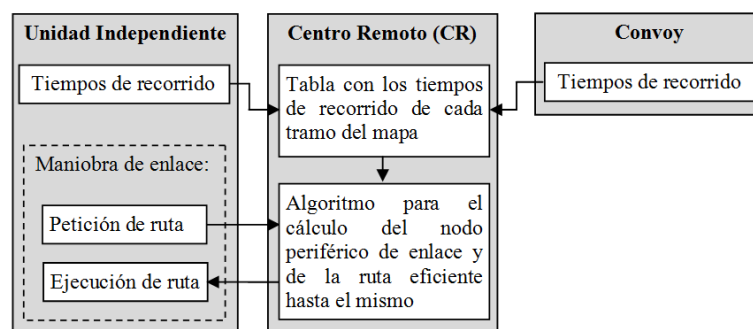


Figura 3.1. Esquema básico de las partes que componen la estrategia de enrutamiento.

### 3.1.1. El escenario de transporte

Como se mencionó en el capítulo introductorio, este trabajo se plantea para brindar soluciones a parte de los objetivos incluidos en el Proyecto *COVE* y tiene como principal motivación la contribución con estrategias válidas a la maniobrabilidad de vehículos autónomos circulando en formación de convoy en escenarios especiales de transporte urbano.

A modo de recordatorio, por entornos especiales urbanos entendemos aquellas áreas donde el tráfico de vehículos convencionales está restringido o prohibido, la velocidad de circulación no supera los 50 Km/h y las trayectorias a través de sus calles suelen presentar fuertes no linealidades. Como ejemplo de este tipo de escenarios se pueden mencionar los núcleos de interés histórico-cultural, parques temáticos o de ocio, entornos portuarios o aeroportuarios, núcleos residenciales, campus universitarios, etc.

En este contexto se justifica la utilización de una flota de vehículos automatizados que, una vez realizada una tarea concreta, pasan a formar parte de un convoy circulando por la periferia y dejando disponibles sus unidades para próximas tareas. A modo de ejemplo, se puede citar el reparto de carga en zonas portuarias, el transporte de personas en núcleos residenciales o de interés turístico, etc. La propuesta idónea sería una flota de vehículos eléctricos automatizados, por su tamaño, eficiencia, y consideración con el medio ambiente (mínima contaminación acústica y nula contaminación de gases y partículas).

El punto de partida de este trabajo está en disponer, en el escenario de transporte, de estos vehículos en uno o varios grupos. Cada grupo está compuesto por varias unidades móviles con autonomía de guiado. Cada unidad de manera independiente realiza las tareas encomendadas y, una vez completadas, cuenta con capacidad para retornar al grupo al que pertenece. En esta tesis se entiende por grupo un conjunto de vehículos en formación de convoy, que está en constante movimiento alrededor de la zona de trabajo, siguiendo una ruta predefinida y cíclica con la función de facilitar la disponibilidad de unidades para las tareas requeridas así como la recuperación de las mismas una vez concluida su misión.

Teniendo en cuenta lo anterior, el sistema propuesto da solución al problema de una unidad móvil independiente que, una vez realizadas las tareas planificadas, se ha de enlazar con su convoy (grupo), que se encuentra circulando alrededor del escenario de transporte.

Para facilitar el análisis, se representa gráficamente un entorno genérico que ayude a entender las estrategias propuestas. Esta representación consiste en un mapa topológico que incluye toda la información vital del entorno sobre el que se aplique la estrategia de enrutamiento, como por ejemplo sus características geográficas, número de calles, longitud de las mismas, y relación entre calles adyacentes a través de intersecciones. La idea de esta representación es simplificar en lo posible la información del entorno real, de modo que el mapa represente las calles mediante líneas o arcos y las intersecciones mediante círculos o nodos, considerando que una calle es el segmento comprendido entre dos nodos adyacentes cualesquiera.

Cada calle del mapa tiene asociado un coste de tiempo estimado a invertir para recorrerla y que dependerá de la longitud de la calle y de las condiciones de tráfico. Este tiempo puede ser fijo si se consideran condiciones de recorrido estáticas, o variable si se consideran condiciones de recorrido dinámicas por las cuales una misma unidad invierte diferentes tiempos en recorrer un mismo tramo en distintas ocasiones, según el momento (día/hora) en que realice la acción. En este capítulo, y como aproximación inicial a la solución global del problema, se analiza el enrutamiento admitiendo que el tiempo de recorrido asociado a cada calle se mantiene invariante en el tiempo a lo largo de la ejecución de la ruta de enlace. Las incertidumbres propias de los tiempos de recorrido características de situaciones reales serán estudiadas en el siguiente capítulo.

En la Figura 3.2 se muestra un ejemplo de mapa topológico correspondiente a una vista parcial del Casco Histórico de la ciudad de Alcalá de Henares (Ciudad Patrimonio de la Humanidad). A la izquierda de la figura se muestra un mapa métrico de esta sección urbana y a la derecha el mapa topológico correspondiente que la simplifica en solo un conjunto de nodos y tramos.



Figura 3.2. Vista parcial del Casco Histórico de Alcalá de Henares. (izda) Mapa métrico. (dcha) Mapa topológico correspondiente al mapa métrico, con las zonas interior y exterior definidas. Los números que aparecen en algunos de los arcos representan el tiempo requerido (en segundos) por un vehículo para atravesar las calles representadas por estos arcos, a una velocidad de recorrido de 24 Km/h.

La estrategia propuesta se basa en gran medida en el análisis de los tiempos de recorrido que tienen que invertir las unidades móviles del entorno para cubrir los distintos tramos del mismo. Estos tramos pueden ser *simples*, situados entre dos nodos adyacentes, o *combinados*, situados entre dos nodos distantes no adyacentes, por lo que un tramo combinado contendrá varios tramos simples consecutivos.

Por otro lado, como se observa en la Figura 3.2, en el problema general se considera que el entorno está dividido en dos zonas diferentes, una exterior y una interior. La zona exterior consiste en una única vía cerrada, que rodea completamente a la zona interior. Un convoy de vehículos circula constantemente por la misma, en recorrido cíclico y de sentido único. Entre las funciones del convoy puede estar la de distribuir las unidades móviles que lo componen

en las zonas del interior del entorno donde sean demandadas. Al terminar su tarea la unidad independiente debe retornar al convoy en el menor tiempo posible de modo que pueda estar disponible para próximas aplicaciones en el entorno de trabajo. La zona interior está formada por el resto de calles del mapa, por tanto en un nodo de esta zona se inicia la maniobra de enlace al convoy.

Dicho lo anterior, antes de cada nueva ejecución de la maniobra de enrutamiento se ha de conocer:

- El nodo interior en el que se encuentre la unidad independiente que desea enlazarse al convoy y que será considerado el nodo inicial de la trayectoria de enlace.
- El nodo periférico en el que está el convoy en el momento de comenzar la maniobra de enlace.
- Los tiempos de recorrido registrados por las diferentes unidades de transporte en los tramos simples. En general los tiempos de recorrido dependerán de la ubicación de los nodos en el escenario de transporte. A modo de ejemplo, en la Figura 3.3, la velocidad de circulación es más baja en los tramos centrales y mayor en los más cercanos a la periferia. Esto significa que la unidad independiente, durante la ruta de enlace puede desarrollar diferentes velocidades, según los tramos atravesados. Esto, además de estar en concordancia a lo que ocurre en entornos urbanos reales.

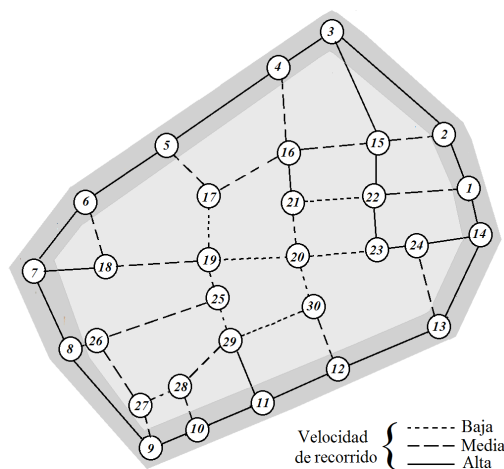


Figura 3.3. Mapa topológico de la Figura 3.2 en el que cada tramo simple tiene asociado un valor (de forma cualitativa: bajo, medio o alto) de la velocidad con que una unidad móvil lo recorrerá.

En todo el análisis de este capítulo, las condiciones iniciales se considerarán conocidas e invariantes mientras dura la maniobra de enlace, aunque cada nueva maniobra puede llevar asociados nuevos nodos iniciales y una distribución diferente de velocidades de circulación en el mapa.

### 3.1.2. El Centro Remoto

El Centro Remoto (*CR*) es un módulo computacional, externo a las unidades de transporte, encargado de coordinar todas las tareas necesarias para lograr el enlace entre una unidad móvil independiente y el convoy, y por tanto tendrá un papel fundamental dentro de la estrategia de enrutamiento propuesta. Las tareas del *CR* se resumen en tres bloques de funciones:

- De almacenamiento de información. Guarda todos los datos concernientes al mapa topológico incluyendo clasificación (periférico o interior) y número de nodos, longitud de cada tramo, así como su correspondiente tiempo de recorrido estimado, fundamentalmente. Toda la información aportada por las experiencias de las distintas unidades que recorren el mapa se almacena aquí para su posible aplicación en la estrategia.
- De procesamiento matemático. Para actualizar la estadística asociada a los tiempos de recorrido de cada tramo y aplicar los algoritmos que proporcionan el nodo óptimo de enlace y la ruta más eficiente hasta el mismo.
- De comunicaciones. El *CR* se encarga de mantener un intercambio de datos constante con todas las unidades de transporte que se encuentran en el mapa.

Mediante estos tres grupos de funciones, el *CR* actúa como agente principal de la estrategia de enrutamiento. A continuación se detallan las tareas principales a realizar por este:

- Actualización de tiempos de recorrido de los tramos simples del mapa. Cuando una unidad móvil, independiente o líder del convoy, atraviesa un tramo simple, envía al *CR* el tiempo experimentado en recorrerlo. Con ello, el *CR* actualiza el registro histórico del tiempo de recorrido en dicho tramo y la estadística asociada, de especial interés cuando se contempla (en el siguiente capítulo de la tesis) incertidumbre en los tiempos de recorrido. De esta forma, cada unidad contribuye con su propia experiencia en la actualización del conocimiento general sobre el escenario de transporte.
- Cálculo de los tiempos de recorrido de los tramos combinados, es decir, aquellos entre nodos no adyacentes. El tiempo que caracteriza un tramo de este tipo se ve afectado por las características de los tramos simples que lo forman. Para calcularlos de forma rápida y precisa se aplican los algoritmos recursivos detallados más adelante.
- Obtención del nodo óptimo de enlace cada vez que es solicitada una maniobra de enlace. Para ello el *CR* debe conocer las ubicaciones actuales del convoy y de la unidad que pretenda enlazarse a él, así como los tiempos estimados de los diferentes tramos simples.



- Cálculo eficiente de la ruta de enlace. El *CR* aplica el algoritmo de enrutamiento diseñado para minimizar su ejecución en tiempo real. El algoritmo obtiene la concatenación de nodos adyacentes que unen la posición actual de la unidad perseguidora con el nodo de enlace obtenido anteriormente.

### **3.1.3. Las unidades de transporte**

La unidad de transporte constituye un término genérico, como mínimo ha de contemplar el registro de los tiempos de recorrido, el control de movimiento autónomo a lo largo de la trayectoria y la comunicación inalámbrica con el *CR* y con el resto de unidades.

Recordando que la idea general parte de que las unidades móviles dejan la formación en convoy para llevar a cabo diferentes tareas de transporte de forma independiente. La labor del convoy es facilitar tanto el acceso de las unidades al punto de aplicación como el retorno a la agrupación concluida la tarea encomendada. En general, la circulación por la periferia es más fluida y es responsabilidad solo de la unidad líder.

Al igual que el *CR*, cada unidad de transporte tiene sus propias funciones. A continuación se detallan las relacionadas con la maniobra de enlace.

#### 3.1.3.1. Tareas llevadas a cabo por el convoy

En la tesis el convoy se define como un conjunto de dos o más unidades de transporte guiadas de forma autónoma siguiendo a un líder, sin enlace físico entre la unidad anterior y la posterior y manteniendo una distancia de seguridad entre ellas. Su ámbito de actuación queda reducido a circular continuamente por la ruta periférica predefinida, cíclica y de sentido único.

La relación entre una unidad con su convoy se puede resumir en una función de seguimiento dentro de la formación y la realización de dos maniobras; una de incorporación de la unidad independiente al convoy y otra de separación del mismo, para moverse también de forma independiente.

La maniobra de incorporación o maniobra de enlace consiste en que una unidad independiente se acople al mismo, bajo condiciones de seguridad para garantizar la integridad de ambas partes. Para ello, la unidad independiente (perseguidora) debe dirigirse desde su ubicación en un nodo interior al nodo periférico en el que se ha de realizar la maniobra de enlace. En este trabajo se considera que el enlace se lleva a cabo con éxito siempre que la unidad perseguidora alcance el nodo de enlace antes de que lo haga el convoy. Por otro lado, se establece que el acoplamiento al convoy se realizará al final del mismo, situándose la unidad enlazada detrás de la última unidad seguidora.

La maniobra de seguimiento, implica que la unidad enlazada va a estar sujeta a los algoritmos propios de cada seguidor [Espinosa *et al.*, 2011] para garantizar el movimiento

estable de la formación (estabilidad de cadena) y la distancia mínima entre unidades; con todo, estos algoritmos no son objeto de esta tesis.

Finalmente la maniobra de separación consiste en lograr que un integrante del convoy, generalmente el último seguidor, lo abandone para realizar una determinada tarea de forma independiente en el escenario de transporte.

En la tesis, se aportan soluciones relacionadas con la maniobra de incorporación al convoy, que por otra parte es la más compleja. Con respecto a esta maniobra, la tarea del convoy consiste en comunicar al *CR* el tiempo invertido en recorrer cada tramo simple y en responder a la solicitud del *CR* sobre la ubicación del mismo. Esta ubicación es necesaria para el cálculo tanto del nodo óptimo de enlace como de la ruta más eficiente hasta el mismo.

### 3.1.3.2. Tareas correspondientes a la unidad perseguidora

Se llama unidad perseguidora a aquella unidad independiente que se dispone a realizar la maniobra de enlace con el convoy. Desde el punto de vista de la estrategia de enrutamiento las funciones de la unidad perseguidora son:

- Enviar su ubicación a requerimiento del *CR* para que este calcule el nodo de enlace y la ruta al mismo.
- Dirigirse al siguiente nodo de la ruta de enlace proporcionada por el *CR*.

## **3.2. Resolución del problema de enrutamiento**

Esta sección estudia en detalle la estrategia de enrutamiento propuesta para resolver el problema planteado en la sección anterior. La estrategia, de forma general, se resuelve en dos etapas vinculadas: 1) encontrar el nodo óptimo de enlace y 2) obtener la mejor ruta que permita a la unidad independiente alcanzar dicho nodo.

Toda la metodología propuesta está basada en el análisis del tiempo de recorrido que caracteriza a cada uno de los tramos simples que componen el mapa de trabajo, es decir el tramo comprendido entre dos nodos adyacentes internos o periféricos. El tiempo de recorrido de un tramo evidentemente es función de su longitud y de la velocidad con que es superado por una unidad de transporte cualquiera y puede considerarse invariante o variante en el tiempo. En este capítulo, como una primera aproximación a la solución global, se considera el caso de velocidad constante.

### **3.2.1. Terminología y definiciones**

Para describir los conceptos asociados a la estrategia de enrutamiento desarrollada se recurre al mapa topológico de la Figura 3.4. Mapa que constituye una abstracción gráfica de un entorno genérico en el que la citada estrategia puede ser aplicada y es trasladable a un entorno urbano real.

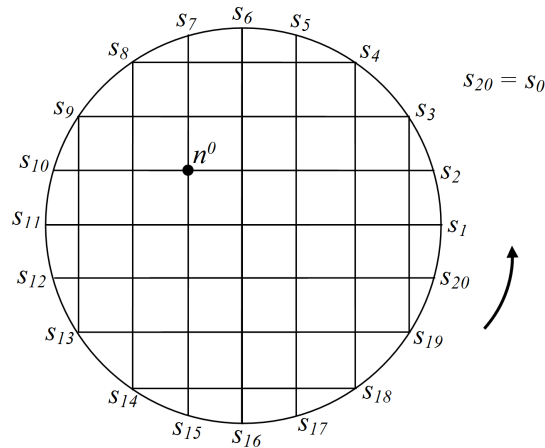


Figura 3.4. Ejemplo de mapa topológico simplificado de un escenario de transporte. Se muestra la ubicación inicial de la unidad perseguidora y la ruta cíclica seguida por el convoy.

Realizando una analogía con la Figura 3.2, aquí también pueden identificarse las dos zonas de interés, la periférica formada por los nodos externos, que a partir de aquí serán caracterizados con la letra  $s$ , y la formada por los nodos interiores, que constituyen el resto y se denotan con la letra  $n$ . A continuación se definirán los siguientes términos:

$n^0$ : Nodo interior en el cual se encuentra la unidad perseguidora en el momento de plantear la maniobra de enlace al convoy.

$s^0$ : Nodo exterior donde está el convoy cuando el perseguidor inicia la maniobra de enlace, desde  $n^0$ .

$P$ : Número total de nodos exteriores, en el ejemplo de la Figura 3.4,  $P = 20$ .

$s_i \in S$ : Nodo periférico cualquiera, siendo  $S = \{s_1, \dots, s_P\}$  el conjunto de todos los nodos periféricos. Puesto que la ruta seguida por el convoy es cíclica,  $s_0 = s_{20}$ ,  $s_1 = s_{21}$ , ...

$s_e$ : Nodo de la periferia elegido por la estrategia de enrutamiento para realizar la maniobra de enlace de la unidad perseguidora al convoy.

$M$ : Número total de nodos interiores del mapa, en el ejemplo de la Figura 3.4,  $M = 37$ .

$n_i \in N$ : Nodo interior cualquiera, siendo  $N = \{n_1, \dots, n_M\}$  el conjunto de todos los nodos interiores.

Con la ayuda de la Figura 3.5 se van a definir los distintos tiempos que son empleados en la estrategia de búsqueda:

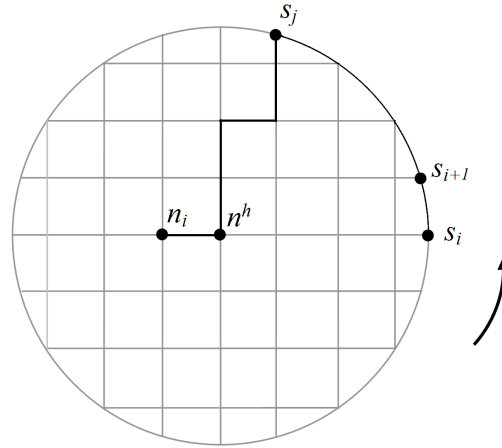


Figura 3.5. Ejemplo de tramos simples entre nodos adyacentes  $(n_i - n^h, s_i - s_{i+1})$ , o tramos combinados entre nodos no adyacentes  $(n_i - s_j, s_i - s_j)$ .

$t(s_i, s_{i+1})$ : Tiempo de recorrido experimentado por una unidad de transporte al viajar desde el nodo periférico  $s_i$  a su sucesor  $s_{i+1}$ .

$T(s_i, s_{i+1})$ : Tiempo de recorrido estimado entre el nodo periférico  $s_i$  y su sucesor  $s_{i+1}$ .

$T(s_i, s_j)$ : Tiempo de recorrido estimado entre los nodos periféricos  $s_i$  y  $s_j$ , no adyacentes.

$t(n_i, n^h)$ : Tiempo de recorrido experimentado por una unidad de transporte por el tramo comprendido entre el nodo interior  $n_i$  y su sucesor  $n^h$ .

$T(n_i, n^h)$ : Tiempo de recorrido estimado entre el nodo interior  $n_i$  y su sucesor  $n^h$ .

$T(n_i, s_j)$ : Tiempo de recorrido estimado entre un nodo interior  $n_i$  y uno periférico  $s_j$ .

Como puede entenderse, los tiempos de recorrido de tramos combinados dependen exclusivamente de los tiempos de los diferentes tramos simples que lo componen. Sin embargo, para viajar de un nodo a otro no adyacente pueden existir múltiples combinaciones de tramos simples, en dependencia de la complejidad del mapa, sin que a priori pueda saberse cuál de ellas tiene el tiempo de recorrido menor. Por ejemplo, en la Figura 3.5, para ir del nodo interior  $n_i$  al periférico  $s_j$  existen varias combinaciones, una de ellas es la mostrada con trazo grueso. Para obtener el tiempo mínimo entre ambos nodos se propone el recurrir a técnicas recursivas que calculan este valor y que finalmente caracterizará el tramo combinado  $(n_i, s_j)$ .

También es importante destacar la diferencia entre tiempo experimentado  $t()$  y tiempo estimado  $T()$ . El primero es el tiempo real registrado por cualquiera de las unidades del escenario de transporte al recorrer un tramo simple del mapa. El segundo constituye una apreciación o estimación del tiempo que empleará una unidad en recorrer el mismo tramo, tiempo que interviene en los procedimientos de cálculo del nodo y ruta óptimos de enlace.

A modo de resumen, en la Figura 3.6 se muestra un diagrama general de la estrategia de enrutamiento donde se observa la relación de funciones entre la unidad perseguidora y el Centro Remoto. Como se ha dicho, en esta primera parte la función del convoy quedará reducida a circular por la periferia siguiendo una ruta predefinida por lo que no se ha incluido en el diagrama.

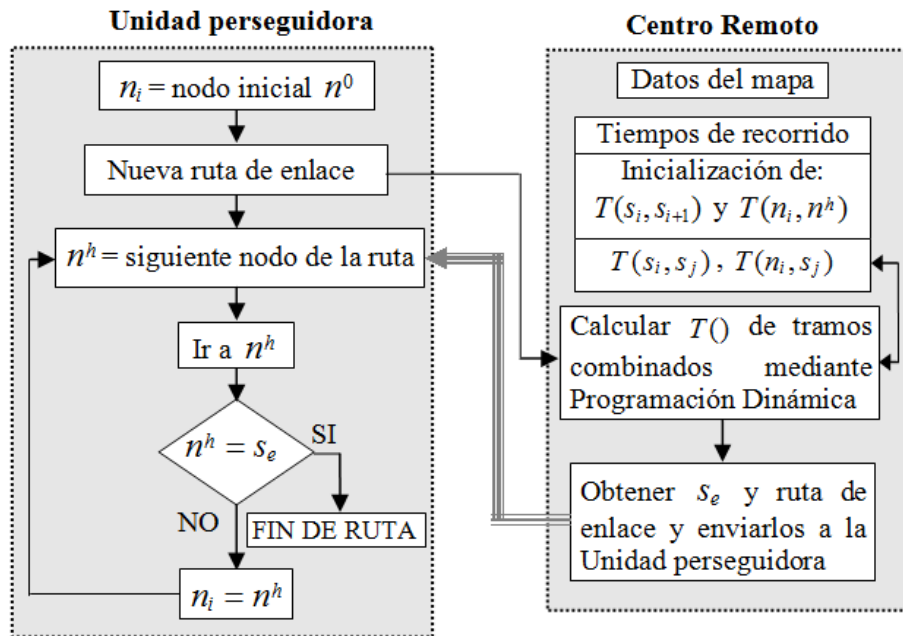


Figura 3.6. Diagrama de bloques general de la estrategia de enrutamiento, con la mayoría de funciones de la unidad perseguidora y el Centro Remoto.

### 3.2.2. Nodo óptimo de enlace

Teniendo en cuenta que la actividad del convoy se limita únicamente a desplazarse continuamente siguiendo la ruta periférica predefinida, el enlace entre una unidad independiente y el mismo debe realizarse necesariamente en uno de los nodos pertenecientes a la periferia.

Se debe recordar que una de las funciones del CR es almacenar todos los tiempos de recorrido del mapa. Inicialmente, cuando no existe experiencia anterior de recorrido, para la ejecución de la estrategia estos valores temporales deben ser inicializados. Esta etapa de inicialización consiste en asignar, a cada tramo simple del mapa, el valor resultante de dividir la longitud del tramo entre la velocidad media de recorrido por el mismo, destacando que el mapa sobre el que se ejecuta la ruta puede incluir varias zonas caracterizadas por diferentes valores de velocidad, siendo generalmente las zonas próximas a la periferia definidas con velocidades mayores que las de zonas más internas, de tráfico menos fluido y más irregular.

Para la estimación del tiempo asociado a tramos combinados, partiendo de los tiempos conocidos de tramos simples, se ha optado por emplear técnicas recursivas de Programación Dinámica.

### 3.2.2.1. Cálculo del tiempo de recorrido entre nodos aplicando Programación Dinámica

La Programación Dinámica (*PD*) fue planteada por Richard Bellman en 1953 [Bellman, 1954], [Bellman, 1957]. Constituye, de forma general, un conjunto de técnicas y algoritmos de optimización, donde la idea central consiste en descomponer un problema complejo en varios subproblemas más sencillos y fáciles de resolver por separado. Para ello se emplea un proceso recursivo en el que se incorporan cada vez más etapas. De esta forma, se combinan soluciones parciales para obtener la solución global del problema original. Aunque es un método similar a los del tipo *Divide y Vencerás* (*Divide-and-Conquer*) [Horowitz *et al.*, 1978], [Kang *et al.*, 2007], [Karimadini *et al.*, 2011], a diferencia de estos, en los que los subproblemas deben ser independientes unos respecto a otros, la *PD* es capaz de brindar soluciones aun cuando los subproblemas presentan algún tipo de solapamiento.

La *PD* se apoya en el Principio de Optimalidad de Bellman: “una política óptima tiene la propiedad de que cualquiera que sea el estado inicial y las decisiones iniciales, las decisiones posteriores deben constituir una política óptima respecto al estado resultante de las primeras decisiones” [Bellman, 1954]. Si se considera que una política es una secuencia de decisiones, y que una política óptima será la más ventajosa entre todas, de acuerdo a algún criterio prefijado, se puede enunciar el principio anterior de la siguiente forma: *dada una secuencia óptima de decisiones, toda subsecuencia de ella es, a su vez, óptima*. Esto es, la solución óptima al problema general se puede encontrar a través de las soluciones óptimas de los subproblemas que lo componen.

Estas técnicas recursivas están presentes en numerosas áreas de la matemática y la cibernética y han sido, por muchos años, exitosamente aplicadas a distintos problemas de enrutamiento [Lee, 1961], [Jefferis *et al.*, 1965], [Willms *et al.*, 2006], [Mainali *et al.*, 2008].

A modo de ejemplo, en [Hu *et al.*, 1997] se aplica *PD* para obtener la ruta óptima de un robot en un entorno con incertidumbres, actualizando los costes del mapa cuando un nuevo obstáculo es encontrado. La *PD* simplifica el proceso permitiendo que todos los nodos afectados por el cambio puedan actualizarse adecuadamente. Algo semejante propone [Cai *et al.*, 2011], mientras [Alton *et al.*, 2008] la aplica a soluciones óptimas en un escenario con múltiples robots dispuestos en diferentes ubicaciones y que requieren encontrarse en un punto común.

[Yu *et al.*, 2009] aplica a su algoritmo iterativo de *PD* la distribución de Boltzmann [Guggenheim, 1955] para optimizar el tráfico en una zona determinada, considerando el volumen del mismo en cada momento. La aplicación de *PD* permite obtener no solo la ruta óptima para cada vehículo, sino un conjunto de rutas candidatas cercanas a la óptima que permiten una mejor distribución del tráfico en la zona, evitando la concentración de vehículos en las mejores zonas.

En [Hashimoto *et al.*, 2008], [Ou *et al.*, 2010] y [Trinh *et al.*, 2011] se aplica *PD* al paradigma específico de enrutamiento de vehículos VRP (*Vehicle Routing Problem*). El VRP consiste en una compilación de problemas de enrutamiento y distribución ampliamente abordado en la

literatura [Golden *et al.*, 2001], [Novoa *et al.*, 2009], [Laporte, 2009], en el que se intenta reducir al mínimo el coste (en tiempo de viaje o distancia recorrida) de un grupo de vehículos que partiendo de un mismo punto deben realizar entregas de mercancías a varios clientes geográficamente distantes.

A su vez, Jula [Jula *et al.*, 2006] investiga un problema del tipo *STSPW* (*Stochastic Traveling Salesman Problem with Time Windows*). Se trata de un caso de distribución de mercancías a varios clientes, con ventanas temporales de entrega, donde los tiempos de recorrido son estocásticos y el objetivo es encontrar la ruta de reparto de menor coste aplicando técnicas recursivas propias de programación dinámica.

A continuación se presenta el procedimiento de cálculo de tiempos de recorrido entre nodos, apoyado en técnicas de *PD*. Para una mejor explicación, el problema se ha dividido en tres partes; cálculo de los tiempos de recorrido de tramos simples (interiores y periféricos), de los tiempos de tramos combinados comprendidos entre dos nodos periféricos, y finalmente de los tiempos de tramos combinados comprendidos entre un nodo interior y uno periférico. A continuación se describe la metodología seguida para cada caso.

#### *Cálculo de los tiempos de recorrido entre nodos periféricos:*

Recordando que cada tramo situado entre dos nodos adyacentes se caracteriza por una velocidad de circulación  $v$ , que varía según la posición geográfica del mismo en el mapa (véase Figura 3.3), y conociendo su longitud  $l$ , dato almacenado en el *CR*, se obtiene fácilmente el tiempo que invertirá una unidad en recorrerlo, siendo  $T_0(s_i, s_{i+1})$  el tiempo de recorrido entre nodos adyacentes periféricos y  $T_0(n_i, n^h)$  el de nodos interiores, si bien la notación anterior es igualmente válida para el caso de que  $n^h$  sea un nodo periférico.

Lo que se desea saber es el tiempo de recorrido  $T(s_i, s_j)$  de todos los posibles tramos combinados de la periferia, comprendidos entre cada par de nodos no adyacentes  $(s_i, s_j)$ ,  $j \neq i+1$ , y recordando que el cálculo se realiza sobre la base de los tiempos de recorrido de los distintos tramos simples que los conformen.

Para lograrlo se aplica el algoritmo de optimización presentado en la Figura 3.7, el cual implementa una expresión recursiva basada en la ecuación de Bellman [Bellman, 1957]. Su aplicación permite obtener los tiempos de recorrido de todos los posibles tramos combinados de la periferia a partir de los tramos simples. Como puede observarse, al comienzo de la ejecución todos los valores de  $T(s_i, s_j)$  requieren una inicialización, que en caso de tiempos entre nodos no adyacentes se realiza a infinito, lográndose reducir la complejidad computacional del proceso recursivo y por tanto aumentar la velocidad de ejecución del algoritmo.

El algoritmo terminará solo cuando se hayan calculado todos los tiempos de recorrido entre nodos periféricos.

<b>Inicialización :</b> <b>desde</b> $i=1$ <b>hasta</b> $P$ <b>desde</b> $j=1$ <b>hasta</b> $P$ <b>si</b> $s_j \notin \text{sucesores}(s_i)$ : $T(s_i, s_j) = \infty$ ; <b>si no</b> : $T(s_i, s_j) = T_o(s_i, s_j)$ ;  <b>Cálculo de</b> $T(s_i, s_j)$ : <b>desde</b> $j=1$ <b>hasta</b> $P$ <b>desde</b> $i=j-1$ <b>hasta</b> $j-P$ ( <b><math>i</math> decrece de uno en uno</b> ) $T(s_i, s_j) = T(s_i, s_{i+1}) + T(s_{i+1}, s_j)$ ; 	<b>Nodos periféricos</b>
--	--------------------------

Figura 3.7. Algoritmo para calcular los tiempos de recorrido estimados entre nodos periféricos no adyacentes.

*Cálculo de tiempos de recorrido entre un nodo interior y otro periférico:*

Para calcular los tiempos de recorrido de tramos interiores del mapa se sigue un procedimiento similar al anterior. En este caso y a diferencia del algoritmo de la Figura 3.7 que analiza todos los nodos de un único conjunto, es decir, pertenecientes a la zona periférica, y donde cada nodo tiene un único sucesor, aquí se relacionan nodos interiores con periféricos. Debido a que cada nodo interior puede tener varios sucesores, se implementa un proceso de selección del nodo sucesor óptimo y que responde a la expresión siguiente, repetida para todos los sucesores  $n^h$  de  $n_i$ :

$$T(n_i, s_j) = \min[T(n_i, n^h) + T(n^h, s_j)]; \quad \forall n_i \in N, s_j \in S, \text{ y } n^h \in \text{sucesores}(n_i) \quad (3.1)$$

El algoritmo propuesto para obtener todos los tiempos  $T(n_i, s_j)$  se muestra en la Figura 3.8.

<b>Inicialización:</b> <b>desde</b> $i=1$ <b>hasta</b> $M$ <b>desde</b> $j=1$ <b>hasta</b> $P$ <b>si</b> $s_j \notin \text{sucesores}(n_i)$ : $T(n_i, s_j) = \infty$ ; <b>si no</b> : $T(n_i, s_j) = T_o(n_i, s_j)$ ;  <b>Cálculo de</b> $T(n_i, s_j)$ : <b>mientras</b> ( <b><math>\text{cambios} \neq 0</math></b> ) $\text{cambios} = 0$ ; <b>desde</b> $i=1$ <b>hasta</b> $M$ <b>desde</b> $j=1$ <b>hasta</b> $P$ $t = T(n_i, s_j)$ ; $T(n_i, s_j) = \min_{\text{sucesores}(n_i)} [T(n_i, n^h) + T(n^h, s_j)]$ ; <b>si</b> $t \neq T(n_i, s_j)$ $\text{cambios} \leftarrow \text{cambios} + 1$ ; 	<b>Nodos interiores</b>
--	-------------------------

Figura 3.8. Algoritmo para calcular los tiempos de recorrido estimados entre un nodo interior y uno periférico.

Como puede observarse, en la valoración del nodo  $n_i$  con respecto a  $s_j$  influye a su vez la valoración que tienen cada uno de sus nodos sucesores  $n^h$ , más el coste de llegar a ellos desde  $n_i$ . Así, al escoger el menor valor de entre todos los posibles, se está asignando al



nodo  $n_i$  la valoración más optimista posible de cuánto costará alcanzar  $s_j$  desde ese nodo. El proceso finalizará una vez todos los tiempos entre nodos interiores y periféricos hayan sido obtenidos y tengan su valor más prometedor.

### 3.2.2.2 Cálculo del nodo óptimo de enlace

Una vez aplicados los algoritmos recursivos presentados anteriormente, el *CR* cuenta con un registro completo de los tiempos estimados de recorrido entre cada par de nodos  $(s_i, s_j)$  y  $(n_i, s_j)$ .

Con esta información, ya se dispone de la mejor estimación de tiempo que empleará una unidad perseguidora en llegar a cada nodo periférico, y del tiempo que empleará el líder del convoy en hacer lo mismo, estando ambas unidades en sus ubicaciones actuales.

Para obtener el nodo óptimo de enlace se realiza una comparación de estos tiempos para cada uno de los nodos periféricos. El nodo bajo estudio, para ser nodo de enlace, deberá cumplir dos condiciones. La primera consiste en que la unidad perseguidora debe alcanzarlo antes de que lo haga el convoy, condición necesaria para garantizar un enlace efectivo, ya que no se contempla que la primera persiga al convoy por la periferia sin haberse producido previamente el enlace al mismo. La segunda condición asegura que el enlace se realice en el menor tiempo posible. Formalmente esto se traduce en la expresión:

$$s_e = \arg \min_i \{T(s^0, s_i) | T(s^0, s_i) > T(n^0, s_i)\}, i \in \{1, \dots, P\} \quad (3.2)$$

Recuérdese que  $n^0$  es la ubicación de la unidad perseguidora en el mapa al solicitar el enlace y  $s^0$  es la ubicación del convoy en el momento de iniciar la maniobra. Puede observarse que (Ec. 3.2) realiza una valoración del tiempo  $T(s^0, s_i)$ , debiéndose a que será precisamente este quien marca el momento en que puede producirse el enlace, precisamente cuando el convoy alcanza  $s_e$ .

El proceso para encontrar el nodo periférico candidato a nodo de enlace comienza analizando el nodo sucesor inmediato al nodo actual del convoy según su sentido de circulación, continuando nodo a nodo hasta detectar al primero que cumpla la condición de (Ec. 3.2). Este proceso repetitivo no supone más cálculo que la propia comparación de tiempos de recorrido, y estos ya han sido obtenidos anteriormente, por lo que no implica ninguna complejidad computacional adicional que dificulte su ejecución en tiempo real.

### 3.2.3. Algoritmo eficiente para cálculo de ruta de enlace

Una vez obtenido el nodo periférico óptimo para realizar el enlace, el siguiente paso es calcular la ruta hasta el mismo mediante un algoritmo de búsqueda eficiente, evitando realizar un estudio sistemático de todos los nodos del mapa hasta dar con el nodo mejor.

En general en los algoritmos de enrutamiento, el proceso de búsqueda requiere que todos los nodos analizados sean almacenados en memoria el tiempo que dure la búsqueda, ya que no

se conoce con anticipación si la ruta que va resultando es la definitiva o será abandonada más adelante por alternativas más prometedoras. Esto contribuye a un coste computacional elevado, más aún cuanto más complejo sea el mapa, variando considerablemente de un método a otro.

El algoritmo propuesto en este trabajo destaca por aportar una solución a este problema, logrando que durante el proceso de búsqueda solo sean elegidos y almacenados aquellos nodos que formen parte de la solución final. De esta forma la propuesta constituye un método óptimo y eficiente, como se demuestra en el análisis matemático del Anexo 1, y que tendrá dos consecuencias inmediatas en la estrategia de enrutamiento:

- Reducción de la memoria empleada durante el proceso de búsqueda. No se almacenan más nodos que los que forman la ruta solución, una ventaja tanto más importante cuanto más extenso y complejo es el mapa.
- Reducción del tiempo de ejecución del algoritmo, haciéndolo idóneo para su aplicación en tiempo real.

A continuación se describe el algoritmo de búsqueda propuesto.

#### 3.2.3.1. Cálculo de la ruta de enlace

Como la mayoría de algoritmos de enrutamiento actuales, el método presentado en este trabajo encuentra la solución mediante la evaluación de los nodos del mapa según el coste que los caracteriza brindando una idea de cuan prometedores pueden o no ser para formar parte de la solución buscada.

El coste de un nodo suele estar en función de la distancia a recorrer para alcanzar el destino, el tiempo empleado en hacerlo o algún otro parámetro definido según el problema a resolver. Por ejemplo, el algoritmo Dijkstra realiza la búsqueda evaluando cada nodo en función del coste que implica alcanzarlo partiendo del nodo inicial, y que es generalmente representado como distancia recorrida. Mientras, el algoritmo "*Primero el mejor*", a diferencia, evalúa cada nodo según el coste temporal de alcanzar el nodo final desde el nodo actual evaluado, sin tener en cuenta lo que se ha invertido en llegar desde el origen al nodo actual. Así, mientras unas soluciones se basan en costes pasados, otras lo hacen sobre estimaciones futuras. El algoritmo A\* combina ambos métodos y su función de evaluación en un nodo incluye el coste de llegar hasta él desde el nodo origen, y el coste estimado de alcanzar el destino a través de él. En otras palabras, evalúa en cada nodo el coste de la ruta origen-destino pasando por dicho nodo.

Aunque estos algoritmos pueden devolver rutas óptimas, la ejecución de la búsqueda demanda el almacenamiento y análisis de un gran número de nodos, dependiendo del método, lo que en grafos muy extensos puede implicar excesivos recursos de memoria y computacionales. Con objeto de minimizar estos recursos se ha diseñado el algoritmo propuesto en la tesis y descrito en la Figura 3.9 [Valdés *et al.*, 2011b].

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. Se parte de <math>n^0</math> y <math>s_e</math> conocidos</li> <li>2. ruta = <math>\{\phi\}</math></li> <li>3. <b>nodo_actual</b> = <math>n^0</math></li> <li>4. Mientras <b>nodo_actual</b> distinto de <math>s_e</math> <ol style="list-style-type: none"> <li>a. <b>nuevo_destino</b> = <math>\arg \min \{T(n^0, n^h) + T(n^h, s_e)\}</math></li> <li>b. ruta = {ruta, <b>nuevo_destino</b> }</li> <li>c. <b>nodo_actual</b> = <b>nuevo_destino</b></li> </ol> </li> </ol> |
|---|

Figura 3.9. Algoritmo de enrutamiento eficiente para obtener la ruta de enlace entre la ubicación actual de la unidad perseguidora y el nodo  $s_e$  previamente obtenido.

El algoritmo basa su funcionamiento en la siguiente estrategia. Se escoge el nodo inicial como primer nodo de la ruta de enlace (**nodo\_actual**). Se generan sus sucesores ( $n^h$ ) eligiendo de entre estos aquel nodo con menor coste y desechando el resto (véase Anexo 1). A continuación, el nodo sucesor elegido pasa a formar parte de la ruta obtenida como segundo elemento (**nuevo\_destino**). Se repite el procedimiento hasta que se alcanza el nodo destino, completando la ruta de enlace.

La elección de nodos se lleva a cabo mediante una función de coste temporal con dos términos que indican el coste pasado y el futuro asociado a la ruta si pasara por el nodo evaluado  $n_i$ :

$$f(n_i) = T(n^0, n_i) + T(n_i, s_e) \quad (3.3)$$

Siendo  $T(n^0, n_i)$  el tiempo empleado en alcanzar el nodo  $n_i$  desde el inicial  $n^0$  y  $T(n_i, s_e)$  el tiempo que se invertirá en ir de  $n_i$  hasta el nodo de enlace  $s_e$ . Recuérdese que los valores  $T(s_i, s_j)$  se calculan mediante el algoritmo recursivo de la Figura 3.8 y se encuentran tabulados en el CR.

La solución propuesta en este trabajo además de ser óptima pues encuentra la mejor ruta, desarrolla una implementación eficiente ya que solo dedica recursos de tiempo y memoria a los nodos que la forman la trayectoria solución. La siguiente sección está dedicada a estudiar ambos aspectos del algoritmo propuesto.

### 3.2.3.2. Eficiencia y optimalidad de la solución planteada

En los algoritmos heurísticos, el número de nodos que se genera como resultado de la búsqueda se ve condicionado por la calidad de la heurística empleada, que también define la optimalidad de sus resultados. A diferencia de algoritmos básicos que realizan un registro sistemático de todos los nodos del mapa hasta dar con el nodo destino, en los más avanzados la heurística ayuda desde el comienzo de la búsqueda a discriminar los nodos que están más alejados de la solución evitando que el algoritmo los evalúe innecesariamente.

Admítase que la búsqueda de la ruta está basada en una función de coste  $f(n)$ , de la forma:

$$f(n) = g(n) + h(n) \quad (3.4)$$

Siendo  $g(n)$  el coste necesario para alcanzar el nodo  $n$  desde el origen, y  $h(n)$  el término heurístico que ofrece una valoración del coste estimado para alcanzar el destino desde  $n$ .

Por otro lado, la optimalidad de la solución está estrechamente vinculada a la heurística empleada, que ha de ser admisible para que el algoritmo asegure una solución óptima. Una heurística se dice admisible [Pearl, 1984] si, en cada nodo analizado, el coste estimado de alcanzar el destino nunca sobre-estima el coste real de alcanzarlo. Esto significa que la heurística empleada es optimista por naturaleza pues asume que el coste de alcanzar el objetivo es menor de lo que realmente puede ser. Una forma muy común de asegurar una heurística admisible, es considerar  $h(n)$  como la distancia euclídea entre el nodo  $n$  y el destino  $s$ , pues al ser recta la línea que une ambos nodos es el camino más corto posible entre ellos (y por tanto el más rápido), cumpliéndose la condición de admisibilidad.

Una de las contribuciones de esta tesis está en minimizar el número de nodos empleados en el proceso de búsqueda y con ello el tiempo y memoria requeridos. Cada vez que se evalúa un nodo y se elige un sucesor para formar parte de la solución, se asegura que este no será descartado en el futuro y que el resto de sucesores son desechados definitivamente. Esto simplifica considerablemente el procedimiento de búsqueda, y aporta una solución idónea para aplicaciones que requieran ejecución en tiempo real. La justificación teórica de la eficiencia del método propuesto se incluye en el Anexo 1 y se demuestra a través del banco de pruebas experimentales presentadas a continuación.

### 3.3. Resultados de simulación

En esta sección se describen dos ejemplos en los que se ha aplicado la estrategia de enrutamiento para demostrar la validez práctica de los algoritmos planteados, empleando para ello dos escenarios diferentes. El primer ejemplo tiene el objetivo de probar cómo la estrategia propuesta obtiene trayectorias óptimas de enlace con el convoy, considerando los tiempos de recorrido estimados de las unidades móviles. Para ello se empleará un mapa sencillo que permite centrar la atención en la obtención de los tiempos de recorrido estimado entre nodos no adyacentes, empleados para alcanzar la mejor solución posible. El segundo ejemplo se orienta a mostrar la eficiencia computacional del algoritmo de búsqueda utilizado, así como su aplicabilidad a situaciones en tiempo real, para lo cual se comparará con el A\*, uno de los algoritmos óptimos más empleados en tareas de enrutamiento. A continuación se detallan ambos ejemplos.

#### Ejemplo 1

Para la realización de este ejemplo se emplea el mapa de la Figura 3.10, que emula una sección urbana con las características mencionadas al comienzo de este capítulo, es decir, con una zona periférica de nodos exteriores y una zona interior. A pesar de la simplicidad del

ejemplo, permite mostrar el desempeño de los algoritmos recursivos empleados, algoritmos que son fácilmente aplicables a mapas más extensos.

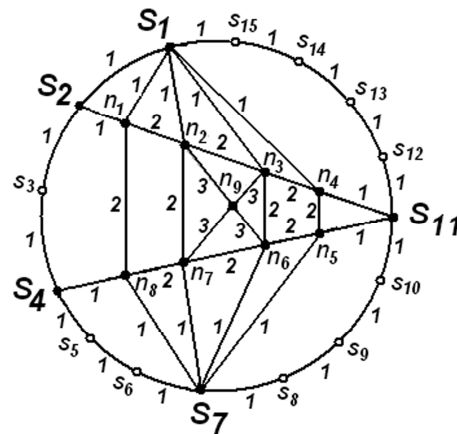


Figura 3.10. Mapa topológico correspondiente a un entorno urbano imaginario. Los números asociados a cada tramo representan los tiempos estimados entre nodos adyacentes.

Como se observa, no todos los nodos periféricos pueden ser alcanzados desde los nodos interiores, como por ejemplo ocurre con  $s_3$ ,  $s_5$  o  $s_6$ , aunque esto no causa ningún problema en la ejecución del algoritmo y la calidad de los resultados.

Para obtener los tiempos estimados de recorrido entre nodos no adyacentes se ejecutaron sobre este mapa los algoritmos recursivos mostrados en las Figuras 3.7 y 3.8. En la Tabla 3.1 se pueden observar los resultados para cada par de nodos periféricos alcanzables desde el interior:

TABLA 3.1. TIEMPOS ESTIMADOS DE RECORRIDO  $T(s_i, s_j)$  ENTRE CADA PAR DE NODOS PERIFÉRICOS ALCANZABLES DEL MAPA MOSTRADO EN FIGURA 3.10.

$T(s_i, s_j)$	$s_j$					
	$s_1$	$s_2$	$s_4$	$s_7$	$s_{11}$	
$s_1$	0	1	3	6	10	
$s_2$	14	0	2	5	9	
$s_i$	$s_4$	12	13	0	3	7
	$s_7$	9	10	12	0	4
$s_{11}$	5	6	8	11	0	

La Tabla 3.2 muestra el coste de ir desde cualquier nodo interior a cualquier nodo periférico alcanzable, y por último, en la Tabla 3.3 se muestran algunas de las rutas óptimas que el algoritmo encuentra para diferentes combinaciones entre la ubicación de una unidad perseguidora y la de un convoy en el momento de solicitar el enlace:

TABLA 3.2. TIEMPOS ESTIMADOS DE RECORRIDO  $T(n_i, s_j)$  ENTRE CADA POSIBLE PAR DE NODOS INTERIOR-PERIFÉRICO ALCANZABLE DEL MAPA MOSTRADO EN FIGURA 3.10.

$T(n_i, s_j)$	$s_j$				
	$s_1$	$s_2$	$s_4$	$s_7$	$s_{11}$
$n_1$	1	1	3	3	7
$n_2$	1	3	5	3	5
$n_3$	1	5	7	3	3
$n_4$	1	7	9	3	1
$n_5$	3	9	7	1	1
$n_6$	3	7	5	1	3
$n_7$	3	5	3	1	5
$n_8$	3	3	1	1	7
$n_9$	4	6	6	4	6

TABLA 3.3. NODO DE ENLACE Y RUTA OBTENIDA PARA DIFERENTES PARES  $(n^0, s^0)$ .

$n^0$	$s^0$	$s_e$	Ruta	$(t_p, t_c)$
$n_9$	$s_1$	$s_7$	$n_9 - n_7 - s_7$	(4,6)
$n_9$	$s_2$	$s_7$	$n_9 - n_7 - s_7$	(4,5)
$n_9$	$s_4$	$s_{11}$	$n_9 - n_3 - n_4 - s_{11}$	(6,7)
$n_9$	$s_7$	$s_1$	$n_9 - n_3 - s_1$	(4,9)
$n_9$	$s_{11}$	$s_1$	$n_9 - n_2 - s_1$	(4,5)

Llegados a este punto, se deben resaltar los siguientes aspectos:

- En este ejemplo la unidad perseguidora siempre partirá del nodo interior  $n_9$ , que es el más alejado de la zona periférica, favoreciendo la diversidad de rutas obtenidas.
- La quinta columna de la Tabla 3.3 muestra los tiempos estimados de recorrido requeridos por la unidad perseguidora ( $t_p$ ) y el convoy ( $t_c$ ) para llegar hasta  $s_e$  que, como se esperaba, es menor en la unidad perseguidora que en el convoy.

En la Figura 3.11, se muestran algunas de las rutas obtenidas para diferentes combinaciones de  $(n^0, s^0)$ :

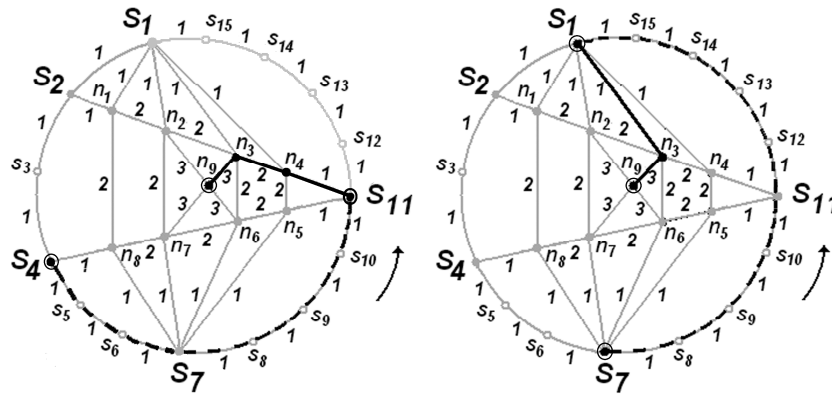


Figura 3.11. Ejemplo de dos soluciones diferentes para las combinaciones  $n^0 = n_9$ ,  $s^0 = s_4$  y  $n^0 = n_9$ ,  $s^0 = s_7$ . La línea discontinua muestra el punto de partida y la ruta seguida por el convoy hasta el nodo de enlace, mientras la línea continua corresponde a la ruta de la unidad perseguidora.

Puede darse el caso de que combinaciones  $n^0$  y  $s^0$  tengan más de una ruta óptima. Si esto sucede, la ruta con menor número de nodos será la elegida como ruta de enlace. La Tabla 3.3 refleja rutas válidas pero no necesariamente únicas conseguidas por la estrategia para unir perseguidor con convoy.

### Ejemplo 2

El segundo ejemplo plantea un escenario de actuación más complejo, representado en la Figura 3.12. Este cuenta con más calles e intersecciones que el mapa del ejemplo anterior y ha sido creado como plataforma para simular la estrategia de enrutamiento mediante la herramienta de simulación Player/Stage [Player/Stage, 2012] ampliamente utilizada en aplicaciones de robótica móvil.

En este ejemplo se pretende mostrar, mediante el entorno indicado, la eficiencia computacional del algoritmo de búsqueda. Algoritmo que es de aplicación a mapas más extensos. Para lograrlo se diseñó un experimento destinado a comparar el método propuesto con el A\*. Los criterios de comparación utilizados fueron dos: el tiempo de ejecución y el número de nodos que se generaron en cada algoritmo durante la búsqueda.

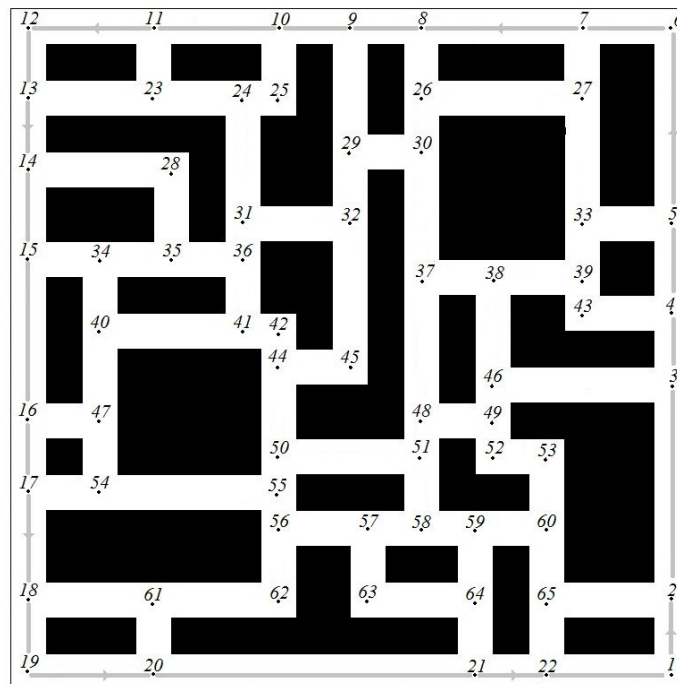


Figura 3.12. Mapa empleado en el Ejemplo 2, construido sobre la plataforma de simulación Player/Stage

Ambos algoritmos fueron programados en lenguaje C y ejecutados en el mismo ordenador, en igualdad de condiciones iniciales sobre el mismo mapa. Se realizaron un total de 20 ensayos combinando diferentes ubicaciones para la unidad perseguidora ( $n^0$ ) y el convoy ( $s^0$ ). Para cada par ( $n^0, s^0$ ), se determinó el nodo de enlace correspondiente utilizando (Ec. 3.2). Seguidamente, ambos algoritmos de búsqueda fueron ejecutados para obtener la ruta hasta dicho nodo, registrándose los tiempos y secuencia de nodos generados en cada ejecución. Se puede comprobar que ambos métodos proporcionan soluciones óptimas.

Debido a la limitada dimensión del ejemplo bajo estudio, los tiempos de ejecución de la búsqueda fueron tan pequeños que podían verse afectados por las fluctuaciones en el rendimiento del sistema operativo sobre el que se ejecutó el simulador. Por ello, se optó por repetir cada ensayo 10 veces y promediar el tiempo de ejecución registrado en cada una de las 10 pruebas. Evidentemente, el número de nodos expandidos de cada caso permaneció constante en las repeticiones. La Tabla 3.4 muestra los resultados correspondientes a los 20 ensayos, mostrando los dos criterios comentados, el número de nodos expandidos en ambos procesos (columnas 5 y 6) y el promedio de los tiempos de ejecución (columnas 7 y 8).

TABLA 3.4. COMPARACIÓN DE NODOS EXPANDIDOS Y TIEMPOS DE EJECUCIÓN DEL ALGORITMO PROPUESTO Y DEL ALGORITMO DE REFERENCIA A\*, EN EL CÁLCULO DE LAS RUTAS PARA DIFERENTES PARES  $(n^0, s^0)$ .

Ensayo	$n^0$	$s^0$	$s_e$	Nodos expandidos por A*	Nodos expandidos por alg. propuesto	Tiempo de búsqueda de A* (ms)	Tiempo de búsqueda de alg. propuesto (ms)
1	24	1	8	10	7	0,0165	0,0103
2	29	1	7	6	5	0,0138	0,0080
3	35	1	8	14	8	0,0229	0,0127
4	37	1	4	5	5	0,0131	0,0078
5	44	1	8	16	7	0,0247	0,0099
6	50	1	9	21	6	0,0406	0,0095
7	54	1	9	20	10	0,0339	0,0120
8	61	1	13	27	13	0,0470	0,0180
9	63	1	13	35	13	0,0594	0,0199
10	65	1	9	19	11	0,0383	0,0147
11	38	4	8	5	5	0,0121	0,0084
12	64	4	15	19	11	0,0392	0,0185
13	32	8	15	7	6	0,0150	0,0104
14	26	8	21	26	9	0,0477	0,0154
15	41	12	16	5	4	0,0124	0,0080
16	27	12	21	21	12	0,0402	0,0201
17	55	16	20	6	5	0,0133	0,0093
18	23	16	3	22	10	0,0379	0,0141
19	56	20	2	12	7	0,0288	0,0137
20	37	20	3	6	4	0,0132	0,0084

Téngase en cuenta que las rutas obtenidas por ambos algoritmos son idénticas en cada ensayo pero, como se indicó en la sección 3.2.3.2, en el nuevo método propuesto solo son analizados los nodos que forman parte de la solución final. Es decir, el algoritmo A\* ha expandido (evaluado) más nodos que los que finalmente forman la ruta de enlace. Esto es así pues estos nodos fueron considerados primeramente como parte de la solución y luego descartados por descubrirse mejores alternativas durante el proceso de búsqueda. Esta es la razón por la que a pesar de que ambos algoritmos alcanzan la misma solución, el número de nodos expandidos es más bajo en el método propuesto en la tesis que en el A\*, implicando un consumo de memoria menor, y tanto más relevante cuanto más complejo sea el mapa.

Esto último se hace evidente si se analizan algunos casos concretos de la tabla. Por ejemplo, si tomamos los ensayos en los que las trayectorias obtenidas han sido relativamente cortas, como en los casos, 2, 11, 15 o 20, puede verse que el número de nodos expandidos es muy parecido o incluso igual en ambas soluciones. Por el contrario, para trayectorias más largas,



la ventaja de un método sobre el otro se hace mucho más palpable. Analizando los casos 8, 9 o 14 se puede apreciar que prácticamente se ha reducido en un 70 % el número de nodos expandidos con la solución propuesta sobre la del A\*. Por otro lado, puede verse una correspondencia directa entre la reducción del número de nodos y el tiempo de ejecución de la búsqueda. Tomando los mismos ejemplos, para los casos 2, 11, 15 y 20, el tiempo se reduce aproximadamente un 40 %, mientras para los casos 8, 9 y 14 se logra alcanzar hasta un 70 % de reducción respecto al tiempo empleado por el A\*.

# 4

## EFECTO DE LA INCERTIDUMBRE EN LOS TIEMPOS DE RECORRIDO

---

Este capítulo completa la estrategia de enrutamiento incluyendo la incertidumbre asociada a los tiempos de recorrido, propia de un entorno real. Estos tiempos, considerados constantes en el capítulo anterior, se modelan como variables aleatorias y se incorporan al cálculo del nodo y ruta de enlace. Además, se propone un procedimiento de evaluación dinámica de la maniobra de enlace, asumiendo un factor de riesgo que modula la probabilidad de éxito de la misma.

### 4.1. Introducción

En el capítulo anterior se plantearon las bases teóricas de la estrategia de enrutamiento propuesta, donde los algoritmos descritos daban solución en un contexto ideal en el que las unidades móviles se desplazaban a una velocidad constante. Esto evidentemente dista de un problema de transporte real, donde las condiciones de las vías y del tráfico, la hora del día, climatología, etc, constituyen importantes fuentes de incertidumbre que repercuten directamente en los tiempos de recorrido y, por tanto, han de ser considerados en la estrategia de enrutamiento.

Por otro lado, en condiciones cambiantes, se hace necesario un procedimiento de actualización de los tiempos de recorrido de cada tramo, almacenados en la base de datos del centro remoto. El objetivo es que la información temporal de cada tramo refleje la experiencia de las unidades que lo atraviesan y sea tenida en cuenta tanto en el cálculo del nodo óptimo como de la ruta más eficiente.

Con todo, el efecto principal de estas incertidumbres es que durante el proceso de búsqueda, la ruta prevista, teóricamente óptima, puede no garantizar el éxito de la maniobra. Es por ello que al proceso de actualización de tiempos se suma otro que comprueba la validez de la ruta seguida cada vez que la unidad perseguidora alcanza un nuevo nodo. Esto permite detectar con antelación si la ruta seguida deja de ser válida y ha de ser recalculada para garantizar un enlace efectivo.

Por último, para acotar la probabilidad de éxito en la realización de la maniobra se introduce el parámetro que, en cierto modo, regula cómo de conservadora o arriesgada se prevé la

solución. A continuación se presenta el análisis de estas incertidumbres y la adaptación a estas de la estrategia de enrutamiento propuesta.

## 4.2. Modelado de la incertidumbre en los tiempos de recorrido

En una situación real de enrutamiento, la incertidumbre inherente al recorrido de los trayectos, de las distintas unidades móviles por el mapa, justifica la naturaleza estocástica del problema. Esto significa que cualquier unidad que atravesase un tramo lo hará con velocidad variante adaptada a las condiciones del escenario de transporte.

En esta tesis, se asume un modelo gaussiano para el tratamiento de dicha incertidumbre. El tiempo de recorrido de cada tramo se considera una variable aleatoria con distribución de probabilidad normal, por ser la forma más extendida de distribución de los procesos estocásticos naturales [Nehmzow, 2006], [Nehmzow, 2008].

Una variable aleatoria  $x$ ,  $-\infty < x < \infty$ , se dice que tiene distribución normal si su función densidad de probabilidad  $f(x)$  responde a la expresión:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.1)$$

Donde  $-\infty < \mu < \infty$  representa la media de  $x$ ,  $\sigma > 0$  su desviación estándar y  $\sigma^2$  su varianza. En la Figura 4.1 se puede ver la forma característica de esta distribución, conocida como Campana de Gauss.

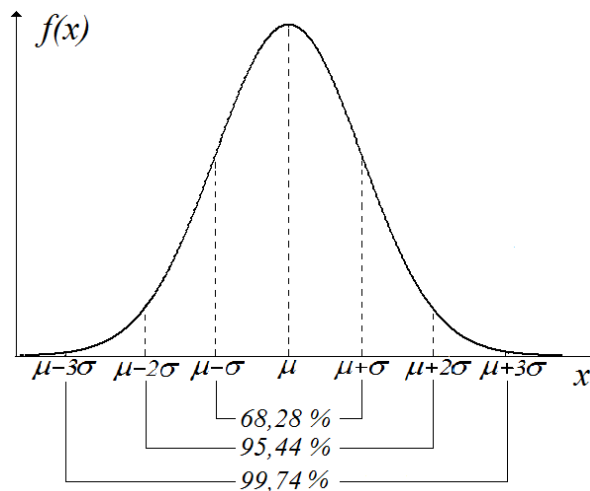


Figura 4.1. Función de densidad de probabilidad de la variable aleatoria gaussiana  $x$ .

Particularizando al problema bajo estudio, es necesario definir los siguientes términos:

$t_p$ : tiempo real que demora la unidad perseguidora, partiendo de su ubicación actual en el nodo interior  $n_i$ , en alcanzar el nodo de enlace  $s_e$ , siguiendo la trayectoria de enlace prevista.

$t_c$ : tiempo real que demora el convoy, desde su ubicación actual  $s_i$ , en llegar a  $s_e$ , siguiendo la ruta periférica.

Si asumimos que el tiempo de recorrido de cada calle se ajusta a una distribución normal, podremos usar esta misma distribución de probabilidad para modelar el tiempo total de recorrido de la ruta. Las funciones de densidad de  $t_p$  y  $t_c$  entonces serán:

$$f(t_p) = \frac{1}{\sqrt{\text{Var}(n_i, s_e)}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t_p - T(n_i, s_e)}{\sqrt{\text{Var}(n_i, s_e)}}\right)^2}, t_p \in \mathfrak{R} \quad (4.2)$$

$$f(t_c) = \frac{1}{\sqrt{\text{Var}(s_i, s_e)}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t_c - T(s_i, s_e)}{\sqrt{\text{Var}(s_i, s_e)}}\right)^2}, t_c \in \mathfrak{R} \quad (4.3)$$

Donde  $T(n_i, s_e)$  y  $T(s_i, s_e)$ , que en el Capítulo 3 se asumían cantidades invariantes, ahora representan la media de los tiempos  $t_p$  y  $t_c$ , mientras  $\text{Var}(n_i, s_e)$  y  $\text{Var}(s_i, s_e)$  son sus varianzas. En la Figura 4.2 se muestran las funciones de densidad de  $t_p$  y  $t_c$ .

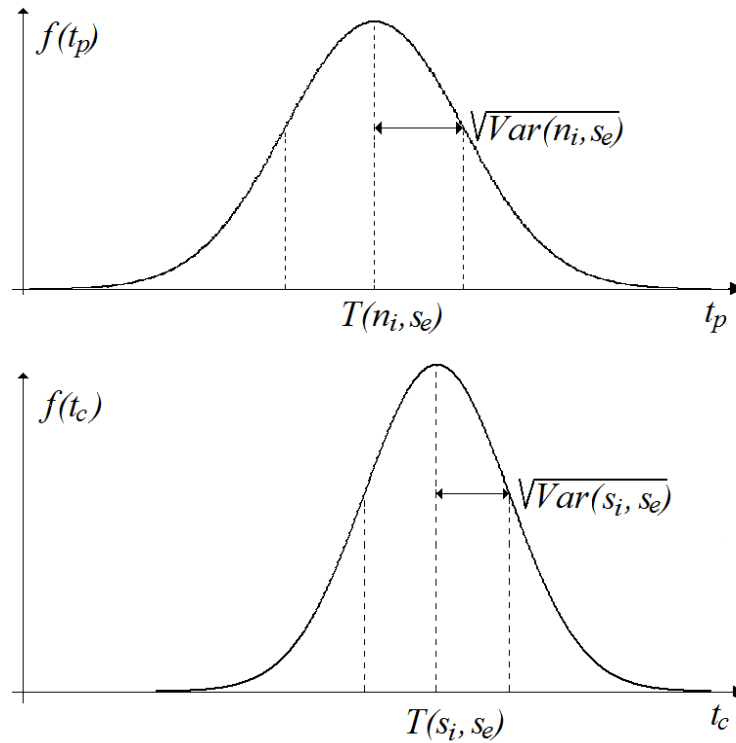


Figura 4.2. Función de densidad de probabilidad de los tiempos de recorrido de unidad seguidora  $t_p$  y convoy  $t_c$ .

Tanto la media  $T()$  como la varianza  $\text{Var}()$  constituyen valores que se actualizan con las sucesivas experiencias de las unidades al pasar por cada tramo simple, y son almacenados en el CR. El procedimiento de actualización se aborda a continuación.

### 4.2.1. Actualización de parámetros de incertidumbre

Actualizar los parámetros de incertidumbre del tiempo de recorrido de un tramo, sea simple o combinado, significa actualizar su media y su varianza. Con ello se pretende que cada nueva experiencia de recorrido de un tramo, sirva para actualizar su modelo de incertidumbre ayudando a mejorar la estimación dinámica del nodo de enlace y de la ruta. A continuación se describe el proceso de actualización estadístico.

#### 4.2.1.1. Actualización de la media y varianza de los tiempos de recorrido entre nodos adyacentes

En el caso de un tramo simple entre un nodo periférico  $s_i$  y su sucesor  $s_{i+1}$ , superado por una unidad móvil, la media  $T(s_i, s_{i+1})$  del tiempo de recorrido estimado que lo caracteriza se actualiza de la siguiente forma:

$$T^n(s_i, s_{i+1}) = \frac{\sum_{k=1}^n t^k(s_i, s_{i+1})}{n} = \frac{\sum_{k=1}^{n-1} t^k(s_i, s_{i+1}) + t^n(s_i, s_{i+1})}{n} = \frac{n-1}{n} T^{n-1}(s_i, s_{i+1}) + \frac{1}{n} t^n(s_i, s_{i+1}) \quad (4.4)$$

Siendo  $n$  el número de veces que el tramo es recorrido,  $t^n(s_i, s_{i+1})$  el tiempo registrado tras la  $n$ -ésima experiencia y  $T^{n-1}(s_i, s_{i+1})$  la media acumulada tras las  $n-1$  experiencias anteriores.

Por su parte, la varianza muestral  $Var(s_i, s_{i+1})$  correspondiente se obtiene según:

$$\begin{aligned} Var^n(s_i, s_{i+1}) &= \frac{\sum_{k=1}^n [t^k(s_i, s_{i+1}) - T^n(s_i, s_{i+1})]^2}{n-1} \\ &= \frac{n-2}{n-1} Var^{n-1}(s_i, s_{i+1}) + (T^{n-1}(s_i, s_{i+1}))^2 - \frac{n}{n-1} (T^n(s_i, s_{i+1}))^2 + \frac{1}{n-1} (t^n(s_i, s_{i+1}))^2 \end{aligned} \quad (4.5)$$

Siendo  $Var^{n-1}(s_i, s_{i+1})$  y  $T^{n-1}(s_i, s_{i+1})$  los valores de varianza y medida acumulados hasta la última actualización  $t^n(s_i, s_{i+1})$ , y  $T^n(s_i, s_{i+1})$  el valor obtenido en (Ec. 4.4).

Para el caso de un tramo simple interior, comprendido entre un nodo interior  $n_i$  y su sucesor  $n^h$ , el procedimiento de actualización sigue una expresión similar:

$$T^n(n_i, n^h) = \frac{\sum_{k=1}^n t^k(n_i, n^h)}{n} = \frac{\sum_{k=1}^{n-1} t^k(n_i, n^h) + t^n(n_i, n^h)}{n} = \frac{n-1}{n} T^{n-1}(n_i, n^h) + \frac{1}{n} t^n(n_i, n^h) \quad (4.6)$$

Y su varianza:

$$\begin{aligned} Var^n(n_i, n^h) &= \frac{\sum_{k=1}^n [t^k(n_i, n^h) - T^n(n_i, n^h)]^2}{n-1} \\ &= \frac{n-2}{n-1} Var^{n-1}(n_i, n^h) + (T^{n-1}(n_i, n^h))^2 - \frac{n}{n-1} (T^n(n_i, n^h))^2 + \frac{1}{n-1} (t^n(n_i, n^h))^2 \end{aligned} \quad (4.7)$$

El CR se encarga de almacenar y actualizar los registros de  $T$  y  $Var$  de cada tramo simple para con estos obtener los valores correspondientes a los tramos combinados, mediante un procedimiento recursivo que será descrito a continuación.

#### 4.2.1.2. Actualización de la media y varianza de los tiempos de recorrido entre nodos no adyacentes

La Figura 4.3 muestra el algoritmo para actualizar la media y la varianza del tiempo de recorrido de tramos combinados periféricos. Básicamente se trata de una adecuación del algoritmo presentado en la Figura 3.7 del capítulo anterior al caso de tiempos con incertidumbres.

<p><b>Inicialización :</b></p> <p><b>desde</b> <math>i = 1</math> <b>hasta</b> <math>P</math></p> <p style="padding-left: 20px;"><b>desde</b> <math>j = 1</math> <b>hasta</b> <math>P</math></p> <p style="padding-left: 40px;"><b>si</b> <math>S_j \notin \text{sucesores}(S_i)</math>: <math>T(S_i, S_j) = \infty</math>, <math>Var(S_i, S_j) = \infty</math>;</p> <p style="padding-left: 40px;"><b>si no</b>: <math>T(S_i, S_j) = f_t(\text{ant}_t(S_i, S_j))</math>;</p> <p style="padding-left: 40px;"><math>Var(S_i, S_j) = f_{var}(\text{ant}_t(S_i, S_j))</math>;</p> <p><b>Actualización de</b> <math>T(S_i, S_j)</math> <b>y</b> <math>Var(S_i, S_j)</math>:</p> <p><b>desde</b> <math>j = 1</math> <b>hasta</b> <math>P</math></p> <p style="padding-left: 20px;"><b>desde</b> <math>i = j - 1</math> <b>hasta</b> <math>j - P</math> (<b><math>i</math> decrece de uno en uno</b>)</p> <p style="padding-left: 40px;"><math>T(S_i, S_j) = T(S_i, S_{i+1}) + T(S_{i+1}, S_j)</math>;</p> <p style="padding-left: 40px;"><math>Var(S_i, S_j) = Var(S_i, S_{i+1}) + Var(S_{i+1}, S_j)</math>;</p>	<p><b>Nodos periféricos</b></p>
---	---------------------------------

Figura 4.3. Algoritmo para calcular media y varianza de los tiempos de recorrido estimados entre nodos periféricos no adyacentes.

Puede observarse que a diferencia del algoritmo de la Figura 3.7, en este la inicialización de los parámetros estadísticos de los tramos simples son función de la experiencia acumulada (tiempos registrados anteriormente, como se indica en las ecuaciones (4.4) y (4.6)) en cada uno de ellos, denotada como  $\text{ant}_t(S_i, S_j)$ . En caso de nodos no consecutivos, una inicialización a infinito responde al mismo objetivo plasmado en la sección 3.2.2.1 del capítulo anterior.

De manera similar, se procede con la estimación de los parámetros estadísticos entre nodos interiores y periféricos, tal y como se muestra en el algoritmo de la Figura 4.4.

La etapa de inicialización sigue el mismo procedimiento que en el caso de nodos periféricos, asignando a medias y varianzas temporales de tramos simples el valor histórico registrado.

En la etapa de actualización la media del tiempo de recorrido de cada tramo combinado  $n_i - s_j$ , se actualiza de forma similar a la mostrada en el algoritmo de la Figura 3.8, mediante programación dinámica. De cada nodo interior, origen de la trayectoria, se elige su sucesor más prometedor o, lo que es igual, el que minimiza el tiempo estimado al nodo periférico

bajo estudio. Para la actualización de la varianza en el tramo  $n_i - s_j$ , se escoge el nodo  $n^h$ , es decir el sucesor de  $n_i$  que minimiza el valor de  $T(n_i, s_j)$  calculado en el paso previo.

<pre> <b>Inicialización:</b> <b>desde</b> i=1 <b>hasta</b> M   <b>desde</b> j=1 <b>hasta</b> P     <b>si</b> <math>s_j \notin \text{sucesores}(n_i)</math>: <math>T(n_i, s_j) = \infty, \text{Var}(n_i, s_j) = \infty</math>;     <b>si no</b>: <math>T(n_i, s_j) = f_t(\text{ant}_t(n_i, s_j))</math>;            <math>\text{Var}(n_i, s_j) = f_{\text{Var}}(\text{ant}_t(n_i, s_j))</math>; <b>Actualización de</b> <math>T(n_i, s_j)</math> <b>y</b> <math>\text{Var}(n_i, s_j)</math>: <b>mientras</b> (<math>\text{cambios} \neq 0</math>)   <math>\text{cambios} = 0</math>;   <b>desde</b> i=1 <b>hasta</b> M     <b>desde</b> j=1 <b>hasta</b> P       <math>t = T(n_i, s_j)</math>;       <math>T(n_i, s_j) = \min_{\text{sucesores}(n_i)} [T(n_i, n^h) + T(n^h, s_j)]</math>;       <math>\text{Var}(n_i, s_j) = \text{Var}(n_i, n^h) + \text{Var}(n^h, s_j)</math>;       <i>donde</i> <math>n^h: \text{sucesores}(n_i)   T(n_i, n^h) + T(n^h, s_j) = T(n_i, s_j)</math>       <b>si</b> <math>t \neq T(n_i, s_j)</math>         <math>\text{cambios} \leftarrow \text{cambios} + 1</math>; </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Nodos interiores</div>
--	---

Figura 4.4. Algoritmo para calcular la media y varianza de los tiempos de recorrido estimados entre cada nodo interior y cada nodo periférico.

### 4.3. Factor de Riesgo

Hasta ahora se ha presentado una propuesta gaussiana de modelado de la incertidumbre en los tiempos de recorrido, y cómo los parámetros que caracterizan estos modelos se actualizan en el CR cada vez que un tramo simple es recorrido por cualquiera de los móviles del escenario de transporte. A su vez estos parámetros son tenidos en cuenta cada vez que se requiere el cálculo del nodo y ruta de enlace.

Dado que ahora se asume que los tiempos de recorrido de cada tramo se comportan como una variable aleatoria, no es posible asegurar que la ruta calculada inicialmente conserve su efectividad durante la ejecución de la misma, de modo que existirá un riesgo de fallo en la maniobra de enlace. Cuanto más lejos del convoy y próximo al perseguidor esté el nodo de enlace propuesto mayor será la probabilidad de éxito pero mayor el tiempo de espera de la unidad perseguidora (solución conservadora). Aquellos casos en los que el enlace se produce antes, pero los tiempos de llegada del convoy y el perseguidor al nodo de enlace son similares, son soluciones más arriesgadas y con una probabilidad de fallo mucho mayor.

Recuérdese que  $t_p$  representa el tiempo experimentado por la unidad perseguidora en alcanzar el nodo de enlace previsto  $s_e$ , y que está dentro de un conjunto de infinitos valores como corresponde al modelo de variable aleatoria gaussiana propuesto en la tesis. Dada la imposibilidad de contemplar el 100% de los posibles casos en el cálculo de la maniobra de enlace se acotará el número de estos a incluir en la solución. Para ello se define  $t_p^*$  como el

límite superior de valores  $t_p$  considerados en el cálculo de la maniobra o, lo que es igual, delimita el porcentaje de posibles valores del tiempo de recorrido del perseguidor considerados en el cálculo de la maniobra de enlace. Este porcentaje puede expresarse de la forma:

$$\theta = F(t_p \leq t_p^*) = \int_{-\infty}^{t_p^*} f(t_p) dt \quad (4.8)$$

siendo  $0 \leq \theta \leq 1$ ,  $\theta = 0$  representa el 0 % de los casos posibles de  $t_p$  y  $\theta = 1$  el 100 %. La figura 4.5 ilustra lo expresado en la (Ec. 4.8).

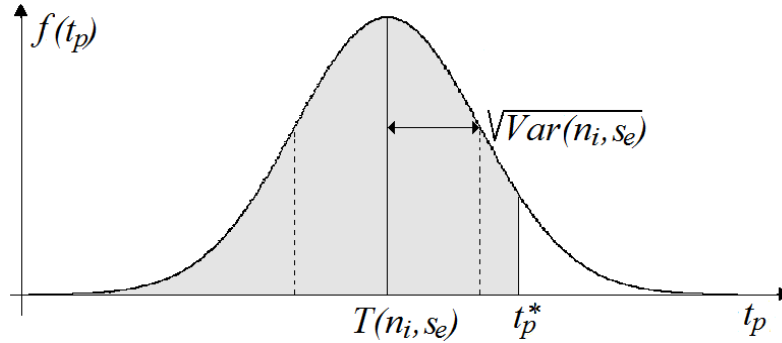


Figura 4.5. Función de densidad mostrando la probabilidad de que se cumpla:  $t_p \leq t_p^*$ .

Con un razonamiento inverso, dada una probabilidad  $\theta$ , se puede calcular el valor de  $t_p^*$  para el que se cumpla que  $t_p \leq t_p^*$  con dicha probabilidad:

$$t_p^* = T(n_i, s_e) + z_p(\theta) \sqrt{Var(n_i, s_e)} \quad (4.9)$$

Donde  $z_p(\theta)$  es la variable aleatoria normalizada resultante de la estandarización de la variable aleatoria  $t_p$ . Para conocer su valor es necesario recurrir a la tabla de probabilidad de la distribución normal estandarizada  $N(0,1)$  [Montgomery *et al.*, 2003]. A modo de ejemplo, si en el cálculo de una maniobra de enlace se considera suficiente el tener en cuenta el 99% de los posibles casos de tiempos de recorrido de la unidad perseguidora ( $\theta = 0.99$ ), resulta  $z_p(0.99) = 2.33$  y por tanto:

$$t_p^* = T(n_i, s_e) + 2.33 \sqrt{Var(n_i, s_e)} \quad (4.10)$$

La Figura 4.6 muestra de forma gráfica varios ejemplos atendiendo al porcentaje de casos contemplados en el tiempo invertido por el perseguidor.

De igual manera se puede acotar la probabilidad de que el convoy alcance el nodo de enlace en un tiempo  $t_c$  menor o igual que un límite  $t_c^*$ .

Al tener en cuenta las incertidumbres en los tiempos de recorrido, el procedimiento para calcular la maniobra de enlace incluye dos etapas:



- a) fijar el porcentaje de casos de tiempos posibles del perseguidor a considerar en la solución. Lo que establece el valor de  $t_p^*$ .
- b) fijar la probabilidad de que el convoy llegue en un tiempo igual o menor que el perseguidor al nodo de enlace, para los casos posibles marcados en el apartado anterior ( $t_c \leq t_p^*$ ), y por tanto que la maniobra sea fallida.

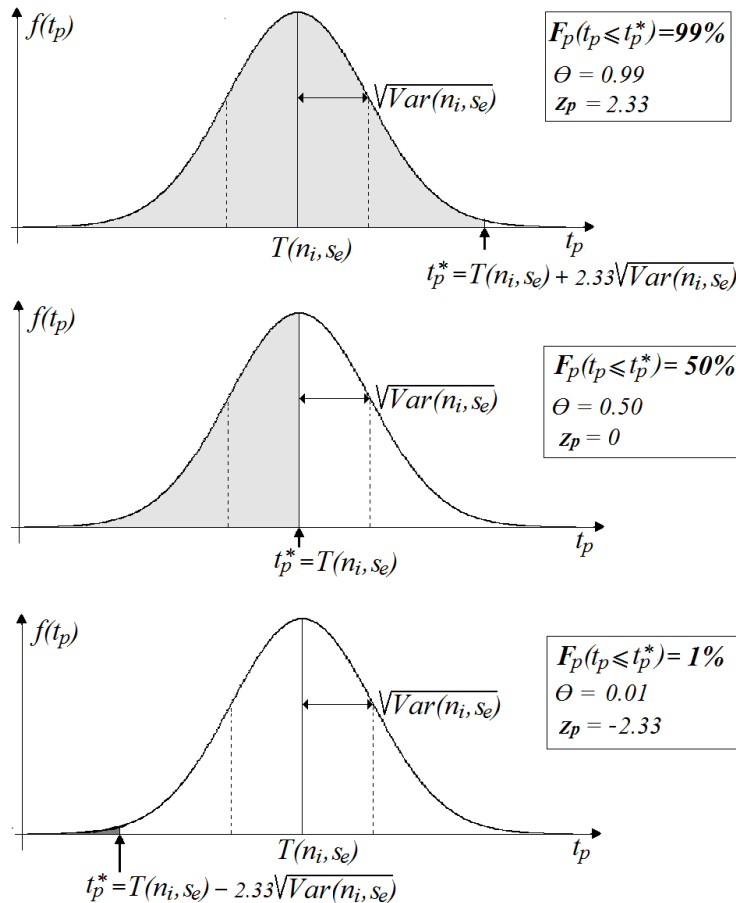


Figura 4.6. Representación gráfica de  $t_p^*$  para distintos valores de  $z_p$  ( $\theta$ ): 2.33, 0 y -2.33.

Para cuantificar la probabilidad de fallo se introduce el concepto factor de riesgo  $\delta$  ( $0 \leq \delta \leq 1$ ). Siendo:

$$\delta = F(t_c \leq t_p^*) = \int_{-\infty}^{t_p^*} f(t_c) dt \tag{4.11}$$

Para evaluar de forma gráfica el significado de  $\delta$  se representan de forma conjunta las funciones de densidad de probabilidad de perseguidor y de convoy, siendo la zona de riesgo el área de  $f(t_c)$  hasta el límite temporal impuesto por  $t_p^*$ . En la Figura 4.7 se indican algunos ejemplos, cuanto mayor es el solape de las funciones de distribución mayor es el riesgo de fallo en la realización de la maniobra.

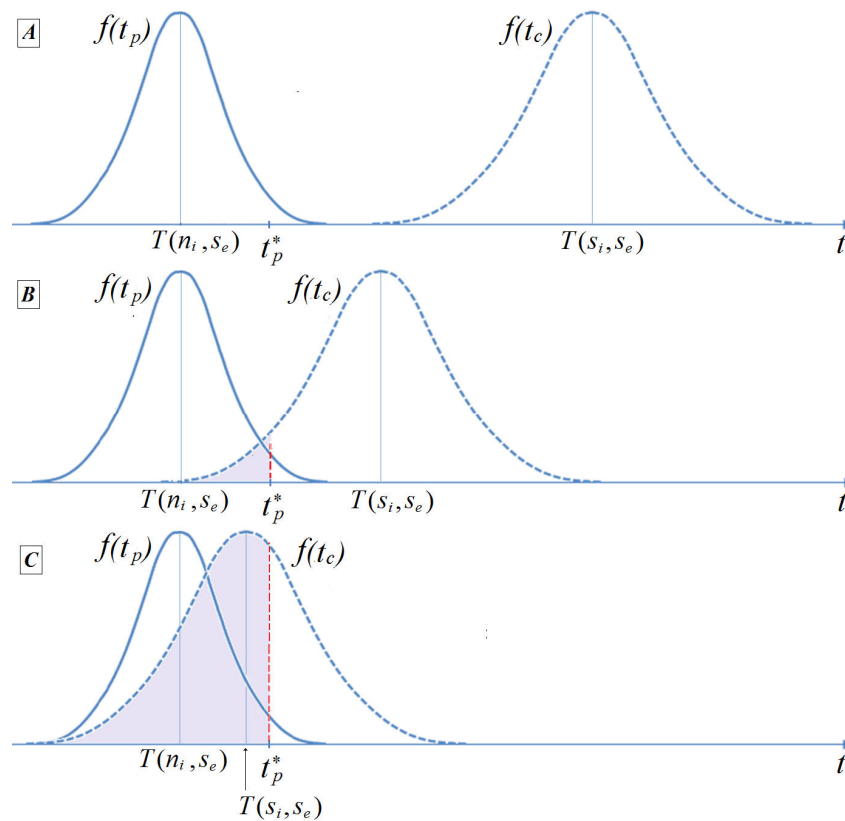


Figura 4.7. Solapamiento de las funciones de densidad de probabilidad de perseguidor y convoy para diferentes casos del riesgo asumido en el cálculo de la maniobra de enlace.

En el caso A ambas curvas están suficientemente alejadas para que el riesgo de fallo sea prácticamente cero. En el caso peor,  $t_p$  máximo y  $t_c$  mínimo, la probabilidad de que el convoy llegue en un tiempo menor o igual que el perseguidor es prácticamente nula.

En el caso B se presenta una situación en la que las funciones de densidad de probabilidad de unidad perseguidora y convoy están solapadas creando una zona de riesgo (área sombreada), con todos los valores de  $t_c$  menores que el límite  $t_p^*$ , lo que corresponde a una probabilidad de fallo  $\delta$  inferior al 50%

Finalmente el caso C muestra un caso de riesgo de fallo superior al 50 %.

#### 4.4. Evaluación dinámica de la maniobra de enlace

En un entorno cambiante, la comprobación de la trayectoria seguida por la unidad perseguidora es obligatoria pues no está garantizada la efectividad de la misma. Hasta ahora se ha mostrado cómo modelar la naturaleza estocástica de los tiempos de recorrido asumiendo un riesgo específico en cada maniobra ejecutada. En esta sección se abordará en primer lugar la incorporación de estas incertidumbres al cálculo del nodo de enlace y posteriormente, se mostrará el procedimiento propuesto para realizar una evaluación dinámica de la ruta y en caso necesario la re-planificación de la misma.

#### 4.4.1. Cálculo del nodo de enlace en función del riesgo asumido

Tal como se planteó en el Capítulo 3, la obtención del nodo  $s_e$  está basada en una comparación de valores conocidos de tiempos de recorrido:  $T(n_i, s_e)$  y  $T(s_i, s_e)$ . Sin embargo al considerar la incertidumbre en los tiempos de recorrido,  $T(n_i, s_e)$  y  $T(s_i, s_e)$  son estimaciones medias de los tiempos  $t_p$  y  $t_c$ ; estimaciones en constante actualización.

Al establecer un riesgo para cada trayectoria de enlace se está induciendo un comportamiento de explotación/exploración en la estrategia de enrutamiento. Explotación en caso de ejecutar rutas más conservadoras y que aseguran un bajo riesgo de fallo, y exploración al obtenerse nuevas alternativas, más arriesgadas y no exploradas que puedan suponer mejores soluciones que las ya existentes.

El nodo de enlace elegido, considerando tiempos con incertidumbres, será aquel que cumpla la siguiente condición:

$$\begin{aligned}
 s_e &= \arg \min_{s_e} \{T(s_i, s_e) / t_c > t_p^*, i = \{1, \dots, P\}\} \\
 &\text{es decir} \\
 s_e &= \arg \min_{s_e} \{T(s_i, s_e) / (T(s_i, s_e) + z_c(\delta)\sqrt{\text{Var}(s_i, s_e)}) > (T(n_j, s_e) + z_p(\theta)\sqrt{\text{Var}(n_j, s_e)})\} \\
 &i = \{1, \dots, P\}, j = \{1, \dots, M\};
 \end{aligned} \tag{4.12}$$

Donde  $z_c(\delta)$  y  $z_p(\theta)$  son obtenidos de las tablas estadísticas dependiendo de los valores del factor de riesgo  $\delta$  y del porcentaje de casos del perseguidor a contemplar en la solución (indicado por  $\theta$ ).

En adelante, el cálculo del nodo de enlace se realiza con un valor fijo de  $\theta = 0.99$ , Esto significa que en el cálculo de  $s_e$  interviene el 99% de los posibles valores de  $t_p$ . Como se resolvió anteriormente:  $z_p(0.99) = 2.33$ , quedando la (Ec. 4.12) de la siguiente forma:

$$s_e = \arg \min_{s_e} \{T(s_i, s_e) / (T(s_i, s_e) + z_c(\delta)\sqrt{\text{Var}(s_i, s_e)}) > (T(n_j, s_e) + 2.33\sqrt{\text{Var}(n_j, s_e)})\} \tag{4.13}$$

Por tanto, el cálculo del nodo de enlace queda en función de un único parámetro: el factor de riesgo. Por ejemplo, admitiendo un riesgo máximo de fallo del 5 % ( $\delta = 0.05$ ), al que corresponde un intervalo de confianza  $z_c(0.05) = -1.65$ , se tiene:

$$s_e = \arg \min_{s_e} \{T(s_i, s_e) / (T(s_i, s_e) - 1.65\sqrt{\text{Var}(s_i, s_e)}) > (T(n_j, s_e) + 2.33\sqrt{\text{Var}(n_j, s_e)})\} \tag{4.14}$$

#### 4.4.2. Re-evaluación de la ruta de enlace y re-planificación

Para explicar las nuevas tareas incorporadas a la estrategia de enrutamiento con incertidumbre nos auxiliaremos de la Figura 4.8, que incluye las nuevas etapas de re-evaluación (1) del nodo objetivo y re-planificación (2) de la ruta seguida.

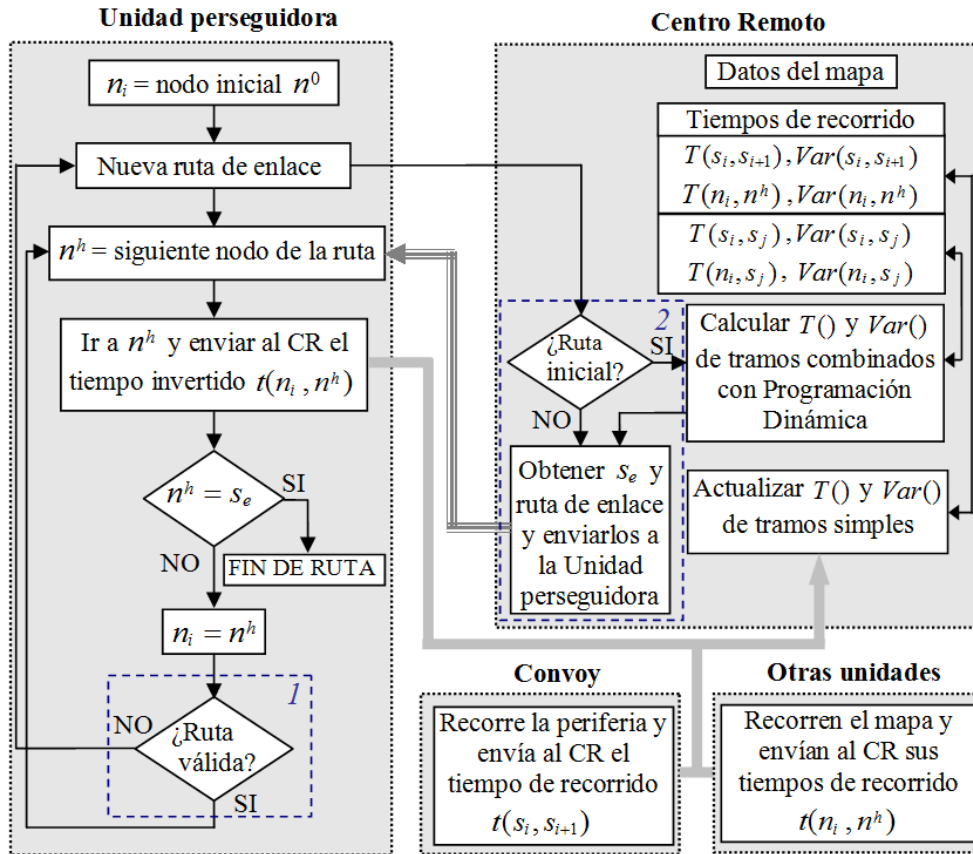


Figura 4.8. Diagrama de bloques de la estrategia de enrutamiento contemplando incertidumbre en los tiempos de recorrido.

La etapa de evaluación dinámica consiste en validar, o no, cada vez que se alcanza un nuevo nodo intermedio, el resultado de (Ec. 4.13), es decir el nodo de enlace  $s_e$ . En caso de cambio del nodo de enlace, se ha de re-planificar la ruta al mismo desde el nodo intermedio actual, con el algoritmo de búsqueda ya presentado (sección 3.2.3.1 de la memoria).

Por último, hay que considerar el siguiente hecho: en el instante en que la unidad perseguidora alcanza un nuevo nodo la ubicación real del convoy no tiene por qué coincidir con uno de los nodos  $s_i$  de su ruta, en general se encontrará en una posición intermedia  $s_x$ . Tomando como referencia la Figura 4.9, que muestra una sección de la ruta periférica con el convoy en  $s_x$ , cuando el perseguidor se encuentra en  $n_i$ , para el cálculo de  $T(s_x, s_i)$  se aplica la velocidad del convoy en el tramo  $(s_{i-1}, s_i)$  a la distancia  $(s_i - s_x)$ .

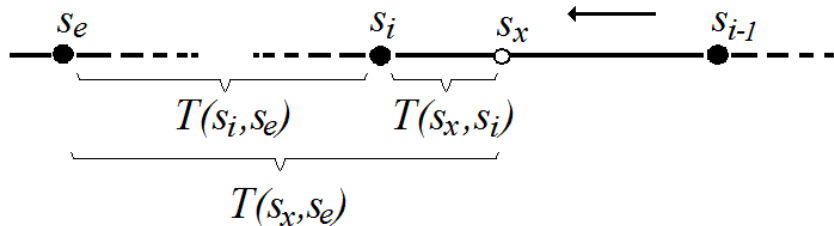


Figura 4.9. Cálculo de  $T(s_x, s_e)$  cuando el convoy se encuentra entre dos nodos periféricos.

Entonces:

$$T(s_x, s_e) = T(s_x, s_i) + T(s_i, s_e) \quad (4.15)$$

$$Var(s_x, s_e) = Var(s_{i-1}, s_i) + Var(s_i, s_e) \quad (4.16)$$

#### 4.5. Resultados de simulación

Esta sección describe los resultados obtenidos mediante simulación y corrobora la efectividad de la estrategia de enrutamiento cuando se contemplan incertidumbres en los tiempos de recorrido de las unidades que intervienen en el escenario de transporte.

Para el experimento se ha diseñado el mapa de la Figura 4.10 que simula un entorno urbano con tiempos de recorrido aleatorios tanto para tramos internos como periféricos. La herramienta de simulación es Player/Stage [Player/Stage, 2012], como ya se hizo en el ejemplo 2 del capítulo 3. Tanto la unidad perseguidora como el líder del convoy se representan mediante dos robots móviles equipados con odometría y sistema láser para el guiado a través del mapa. En cada ensayo la unidad perseguidora comenzará en un nodo interior y deberá calcular el nodo y ruta de enlace con un convoy que se mueve por la periferia. Además, cada vez que alcanza un nuevo nodo durante su recorrido, debe evaluar si procede la re-evaluación del objetivo y, en tal caso, re-planificación de la trayectoria, siguiendo el proceso hasta que la maniobra sea dada por exitosa o fallida.

A efectos de simular el problema de las incertidumbres en los tiempos de recorrido, cada tramo del mapa será recorrido con una velocidad aleatoria  $v$ , la cual sigue una distribución de probabilidad normal, con una media  $\mu_v$  y una varianza  $\sigma_v^2$ . Para ello, se diseñó un generador de números aleatorios que implementa la forma polar de la transformación de Box-Muller [Box *et al.*, 1958], cuyas entradas son la media  $\mu_v$  y varianza  $\sigma_v^2$ , propias del tramo simple que vaya a recorrer la unidad y definidas con antelación; y cuyas salidas son la velocidad  $v$  a aplicar a la unidad móvil en su recorrido.

Para simular un caso real, se ha optado por dividir el mapa en tres zonas: interior, exterior e intermedia, cada una con sus propios parámetros estadísticos. Así, los tramos simples pertenecientes a la zona interior se caracterizarán por velocidades más bajas pero con mayor varianza, en correspondencia con un tráfico lento e irregular, mientras que los de la zona exterior o periférica se caracterizan por velocidades más altas y regulares. La tercera zona agrupa tramos con valores intermedios entre las zonas anteriores. Para el ejemplo en cuestión, la distribución de medias y varianzas es la siguiente:

- Tramos de la zona central del mapa:  $\mu_v = 0.4 \text{ m/s}$ ,  $\sigma_v^2 = 0.04 \text{ (m/s)}^2$
- Tramos de la zona intermedia:  $\mu_v = 0.6 \text{ m/s}$ ,  $\sigma_v^2 = 0.01 \text{ (m/s)}^2$
- Tramos de la zona periférica:  $\mu_v = 0.8 \text{ m/s}$ ,  $\sigma_v^2 = 0.0025 \text{ (m/s)}^2$

Para comparar el desempeño de la estrategia de enrutamiento asumiendo riesgos diferentes, se han realizado tres pruebas con múltiples ensayos en cada una. Se ha decidido mostrar los 60 primeros pues es un número suficiente para extraer conclusiones válidas. En todas las simulaciones la unidad perseguidora parte del nodo interior 55 mientras que el convoy lo hace del nodo periférico 1, ambos señalados con trazo azul y rojo respectivamente en la Figura 4.10. La trayectoria del convoy a lo largo de la periferia está predefinida y tiene sentido anti-horario.

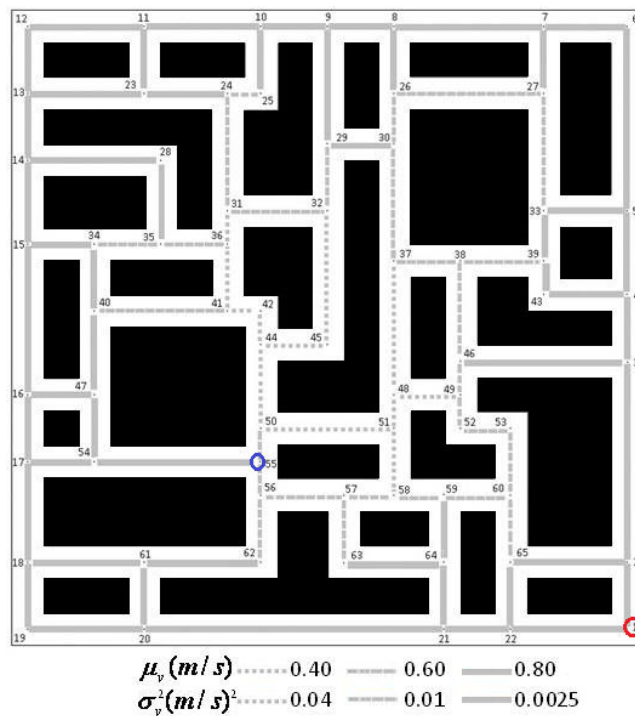


Figura 4.10. Distribución de medias y varianzas de los tramos para diferentes zonas del mapa bajo estudio.

Todas las pruebas parten de las mismas condiciones iniciales y al comenzar cada batería de ensayos se va a asumir que el mapa no ha sido recorrido antes. Por tanto, ha sido necesario un procedimiento de inicialización de medias y varianzas de los tiempos de recorrido de los tramos simples. Las medias  $T(s_i, s_{i+1})$  y  $T(n_i, n^h)$  se inicializan en función de la longitud del tramo y de la velocidad media estimada de circulación  $\mu_v$  de cada tramo; y sus varianzas se inicializan a un valor superior (10 veces mayor) a las características  $\sigma_v^2$  de cada tramo. Todas estas variables se irán actualizando cada nueva experiencia mediante las (Ec. 4.4), (Ec. 4.5), (Ec. 4.6) y (Ec. 4.7).

Cada prueba se caracteriza por el valor de riesgo de fallo asumido, un 5 % en la primera, un 50 % en la segunda y un 95 % en la tercera. A continuación se describen los resultados y conclusiones obtenidas en cada una de ellas.

**4.5.1. Prueba 1. Maniobra de enlace con factor de riesgo del 5 %**

En la Tabla 4.1 se muestran los resultados de la maniobra de enlace cuando se contempla un riesgo de fallo del 5 % ( $\delta = 0.05$ ). La primera fila indica el número de los ensayos, mientras la segunda muestra el nodo de enlace  $s_e$  obtenido inicialmente en cada ensayo. Las filas de la tercera a la séptima muestran el nodo de enlace resultante de las sucesivas re-planificaciones llevadas a cabo en cada ensayo (con un máximo de hasta 4 re-planificaciones en los ensayos 3, 21 y 29). Finalmente la octava fila indica si el enlace tuvo éxito (OK) o fracasó (NO). A modo de ejemplo, en el ensayo 20, puede observarse que inicialmente  $s_e = 14$ , sin embargo, tras varias re-planificaciones a lo largo de la maniobra resultan los siguientes valores del nodo de enlace:  $s_e = 13$  y  $s_e = 11$ .

TABLA 4.1. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 1, ASUMIENDO UN RIESGO DE FALLO DEL 5 %.

Nº Sim	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Nodo $s_e$	9	10	13	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
	13	11	11									13	13		13					13	13			13	13		13		13		
	14	9	13									11	10			11				11	14			11	14		14		11		
		14	11									13					10				13			10						13	
			13																			10									11
Enlace	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Nº Sim	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
Nodo $s_e$	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
	13				13	13			13	13		13	13	13			13	13	13				13				13	13	13	13	
	11				10	14			14	11		11	11	14				14	14	14				11				11	14	11	11
	10										10	10																13		9	10
Enlace	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Del análisis de esta prueba se desprenden las siguientes conclusiones:

1. El valor inicial del nodo de enlace  $s_e$  (segunda fila) ha evolucionado con el número de ensayos, de obtenerse valores iniciales más próximo a la ubicación del convoy ( $s_e = 9, 10$ ) a nodos más conservadores ( $s_e = 13, 14$ ). Esta evolución se debe a la actualización de los valores estadísticos, desde los aplicados en la etapa de inicialización, tras las sucesivas ejecuciones. Finalmente se tiende a un valor estable del nodo de enlace previsto ( $s_e = 14$ ) al inicio de cada maniobra.
2. En todos los casos el objetivo fue conseguido, aunque al aumentar el número de ensayos podrían aparecer fallos aislados. Sin embargo, al tratarse de una estrategia donde la ruta puede ser corregida en tiempo real, el porcentaje de fallos nunca superará el valor del factor de riesgo asumido (5%).

3. Un factor de riesgo bajo implica maniobras más conservadoras (a nodos  $s_e$  más distantes del convoy), y por tanto se requieren menos re-evaluaciones de las soluciones previstas. La evolución en el número de re-planificaciones es mostrada en la Figura 4.12, el valor medio de las mismas en esta prueba fue de 1.2 re-planificaciones por ensayo.

#### 4.5.2. Prueba 2. Maniobra de enlace con factor de riesgo del 50 %

En esta segunda prueba se evalúa la estrategia de enrutamiento cuando el riesgo de fallo asumido es del 50 % ( $\delta = 0.50$ ). Para ello se repite el experimento con las mismas condiciones iniciales de la Prueba 1, a excepción del factor de riesgo empleado. Los resultados son mostrados en la tabla 4.2, que mantiene la misma estructura de la tabla anterior.

TABLA 4.2. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 2, ASUMIENDO UN RIESGO DE FALLO DEL 50 %.

Nº Sim	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Nodo $s_e$	9	10	11	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
	14	11	10	13					13	13	13	13	13		13		13	13	13	13	13	11	13	13	13	13	13	13	13	
	16	10	11	21					9	9	10	9	9		10		10	10	11	11	10	10	3	14	14	14	14	11	14	11
	20	11		2					8					14						10	10			2						10
		10																												
Enlace	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK

Nº Sim	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Nodo $s_e$	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
	13	13	13	13	13	13	13	13	13	13	13	13	13	13		13	13	13	13	13	13	11	11	13	13	13	13	13	13	
	11	11	11	11	11	10	11	10	11		11	14	11	11	11		11	11			11	10	13	11	10	14	11	10	11	14
	10		10	10	10		10		10		10		10	10	10						10	10					10		13	
				11																										
Enlace	OK	OK	OK	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	NO	OK	OK	OK	OK	OK

De los resultados de la Prueba 2 se derivan las siguientes conclusiones:

1. La consecuencia directa de un incremento en el riesgo asumido es el aumento en el número de re-planificaciones durante la maniobra de enlace, en esta prueba en particular la unidad perseguidora tuvo que cambiar de nodo de enlace un 78 % de veces más que en la anterior (en total 128 contra 72 de la Prueba 1). En este caso el promedio de re-planificaciones por ensayo alcanzó un 2.1, contra un 1.2 de la anterior.
2. El número de fallos aumentó al 8.3 % (5 en 60 ensayos), en este caso muy por debajo del estimado (50 %) si no se permitiesen re-planificaciones.
3. La Figura 4.11 incluye dos capturas de pantalla tomadas durante los experimentos 32 (Fig. 4.11. A) y 37 (Fig. 4.11.B) de la Prueba 2. En estas se muestran las trayectorias seguidas y la disposición final de la unidad perseguidora y del convoy. Los nodos



interiores donde la unidad perseguidora realizó las re-planificaciones están marcados con un cuadrado, siendo el número en su interior el nodo de enlace elegido en cada una de ellas. En el caso A, se puede observar que la maniobra de enlace ha sido exitosa, mientras que en el caso B, fallida. También se observa que la trayectoria seguida por la unidad perseguidora hasta el penúltimo nodo interior (nodo 24) es la misma en ambos casos. Sin embargo, el perseguidor alcanza dicho nodo con una ventaja temporal sobre el convoy que es mayor en el caso B que en el caso A, lo que unido al riesgo de la prueba hace que la unidad perseguidora decida ser más agresiva y optar por un nodo de enlace como el  $s_e = 10$ . El hecho de recorrer el último tramo a una velocidad menor de la esperada provoca el fallo de la maniobra de enlace. Evidentemente, la imposibilidad de la unidad perseguidora de re-planificar la ruta una vez se encuentra en la parte final del trayecto condiciona el resultado fallido de la maniobra de enlace.

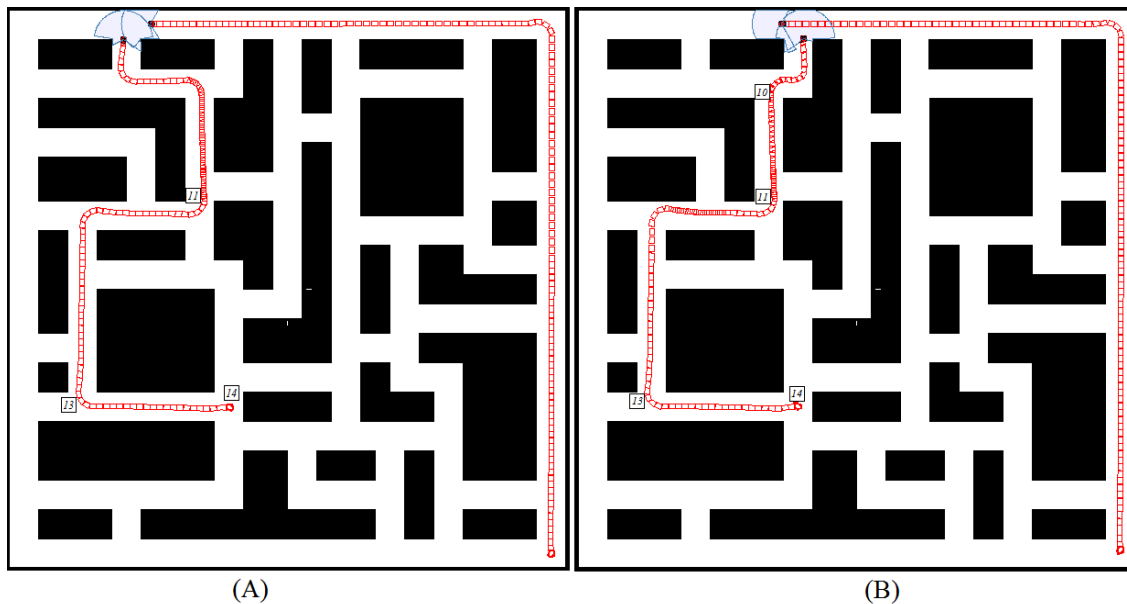


Figura 4.11. Imágenes tomadas durante las dos simulaciones de Player/Stage donde ambas maniobras de enlace tienen el mismo riesgo del 50 %. (A) muestra una maniobra exitosa, mientras la mostrada en (B) es fallida.

#### 4.5.3. Prueba 3. Maniobra de enlace con factor de riesgo del 95 %

Por último, se realizó una tercera prueba asumiendo un riesgo de fallo del 95 % ( $\delta = 0.95$ ), manteniendo las mismas condiciones iniciales que en las pruebas anteriores. Los resultados se muestran en la Tabla 4.3.

TABLA 4.3. RESULTADOS DE SIMULACIÓN DE LA PRUEBA 3, ASUMIENDO UN RIESGO DE FALLO DEL 95 %.

Nº Sim	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Nodo $s_e$	9	8	9	10	11	11	11	11	10	10	11	11	13	13	13	11	13	13	13	13	13	13	13	13	13	13	13	13	13	13
	8	9	13	11	10	10	13	10	11	11	13	13	11	11	10	13	11	14	10	11	11	11	11	11	11	14	14	14	10	11
		15	14	10	9	11	10		10	10	10	10	10	13		11	13	13		10	10	10	13	10	13	13	11	13		10
		16	21	11		10					13		11		11		13	21			13		11		21	11	13	11		
		20	2														11	2							2	10	10	10		
		17															13	3							21					
Enlace	OK	OK	NO	OK	NO	OK	OK	OK	OK	NO	OK	OK	OK	NO	OK	OK	NO	OK	OK	OK	OK	OK	NO	OK	NO	OK	OK	OK	OK	OK

Nº Sim	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Nodo $s_e$	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
	11	10	11	10	11	11	11	11	11	11	11	11	11	11	11	10	11	10	11	11	11	11	11	10	11	11	11	11	11	11
	10	11	13		10	10	13	13	10	10	13	10	10	10	13		10		10	13	10	10	10		10	13	10	13	10	13
		10	10		11		11	11			11		11		3				11	11						10	11	11	13	11
		11						10			13				2				10								10	10		10
		10																		13										13
Enlace	OK	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK	OK	OK	NO	OK	OK	NO	OK	OK	OK	OK	OK	OK	OK	NO	OK	OK	NO	OK

De esta prueba se pueden extraer las siguientes conclusiones:

1. El nodo de enlace inicialmente elegido es menos conservador que en las pruebas anteriores, resultando nodos más cercanos a la ubicación inicial del convoy ( $s_e = 10, 11, 13$ ). Además, en el 73 % de todos los ensayos en los que hubo re-planificación el  $s_e$  final es más cercano a la posición de origen del convoy que el  $s_e$  inicialmente previsto.
2. Se incrementa el número de re-planificaciones. En esta prueba en particular la unidad perseguidora tuvo que cambiar de nodo de enlace un 28 % de veces más que en la prueba anterior (en total 164 contra 128 de la Prueba 2). Obviamente el promedio de re-planificaciones por ruta también aumentó, siendo de un 2.73 en esta prueba contra un 2.1 de la anterior. A modo de comparativa, la Figura 4.12 muestra la evolución del número de re-planificaciones en cada una de las pruebas realizadas. Además se señalan los distintos fallos para cada prueba (recuérdese que en la Prueba 1 no hubo ninguno).
3. Se incrementa el número de maniobras fallidas, alcanzando un 20 % (12 fallos en 60 ensayos). Evidentemente el número de fallos sería muy superior sin la posibilidad de corregir la ruta a tiempo.

La Figura 4.13 muestra el total de veces que cada nodo de enlace es elegido a lo largo de los 60 ensayos de cada prueba, demostrándose que un factor de riesgo mayor induce a la unidad perseguidora a elegir nodos  $s_e$  menos conservadores. Esto puede deducirse de la siguiente observación: el nodo  $s_e = 10$  fue seleccionado 37 veces en la Prueba 3 (riesgo del 95 %) y tan solo 9 en la Prueba 1 (riesgo del 5%). Por el contrario, un nodo más conservador como el  $s_e = 14$  es elegido 43 veces en la Prueba 1, 16 veces en la Prueba 2 (riesgo del 50 %) y en ninguna ocasión en la Prueba 3.

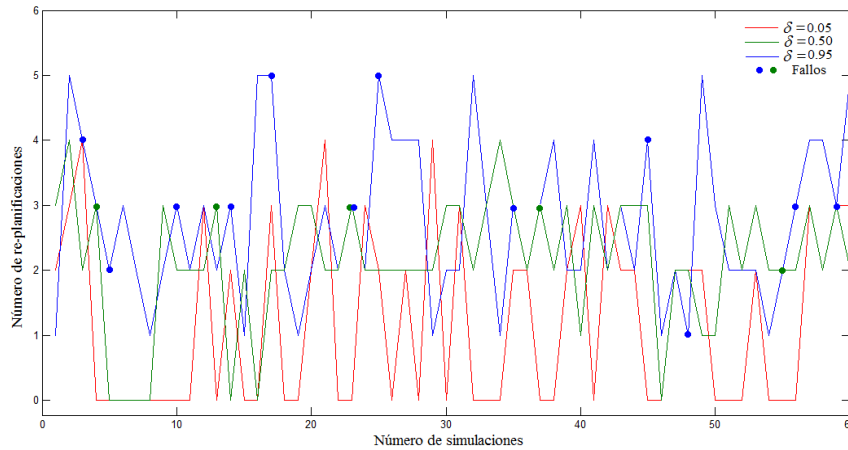


Figura 4.12. Número de re-planificaciones realizadas por la unidad perseguidora en las tres pruebas.

La conclusión general, tras los ensayos de las tres pruebas es la siguiente: la propuesta de factores de riesgo más altos implica maniobras con mayor probabilidad de fallo pero con tiempos de enlace al convoy más bajos en caso de éxito.

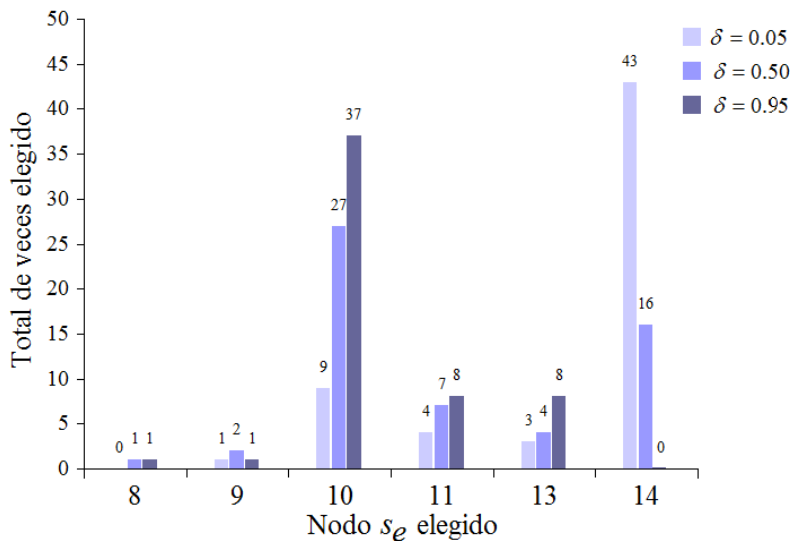


Figura 4.13. Número de veces que cada nodo periférico  $s_e$  es elegido como nodo de enlace durante las simulaciones realizadas en las Pruebas 1 (riesgo del 5 %), 2 (riesgo del 50 %) y 3 (riesgo del 95 %).

Por último hay que tener en cuenta que los resultados obtenidos en las pruebas realizadas están condicionados por la limitada complejidad del mapa utilizado.

#### 4.6. Alternativas para el modelado de incertidumbres en los tiempos de recorrido

Aunque en la parte fundamental del trabajo realizado en la tesis se ha utilizado una distribución de probabilidad gaussiana para modelar el comportamiento aleatorio de los tiempos de recorrido de las unidades de transporte, existen otras funciones de distribución analizadas en diversos estudios de tráfico y problemas de enrutamiento.

Diversos estudios relacionados con los tiempos de recorrido en circuitos urbanos asumen la distribución normal en sus modelos [Xu *et al.*, 2008], [Rakha *et al.*, 2006]. En [Berry *et al.*, 1951] se justifica que esta es la mejor distribución cuando la señalización y otras condiciones de tráfico imponen retrasos importantes. También en el contexto robótico es una distribución ampliamente utilizada [Nehmzov, 2006], [Nehmzov, 2008]. La principal ventaja de modelar la incertidumbre en el recorrido con distribuciones gaussianas es que, asumiendo este tipo de distribución para los tramos independientes entre nodos consecutivos, un tramo combinado puede analizarse como una concatenación de funciones gaussianas y, por tanto, responde también a una distribución gaussiana.

Sin embargo, otros trabajos realizados en situaciones habituales de tráfico han demostrado que los tiempos de recorrido tienden a una distribución de probabilidad cuya curva está sesgada a la derecha [Sun *et al.*, 2003], [Kwon *et al.*, 2000]. Esto sucede por ejemplo cuando el número de vehículos excede la capacidad de la vía [Kharoufeh *et al.*, 2004]. Este comportamiento asimétrico de la distribución de los tiempos de recorrido ha llevado a que diferentes investigaciones sobre el modelado del tráfico asuman funciones no gaussianas. Entre las más utilizadas están la Log-normal, la Gamma y la Exponencial.

#### 4.6.1. Función de distribución Log-normal

Se dice que una variable aleatoria  $x > 0$  tiene distribución Log-normal cuando su logaritmo  $\ln(x)$  presenta una distribución normal  $N(\mu, \sigma^2)$ . Su función de densidad de probabilidad  $f(x)$  responde a la siguiente expresión [Montgomery *et al.*, 2003]:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\mu}{\sigma}\right)^2}, 0 < x < \infty \quad (4.17)$$

Siendo  $\mu$  y  $\sigma^2$  la media y varianza respectivas en la distribución normal de  $\ln(x)$ , mientras que  $\theta = e^{\mu + \sigma^2/2}$  y  $\omega^2 = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$  lo serán en la distribución Log-normal de  $x$  (Ec. 4.17). Para valores medios o altos de  $\sigma$  esta distribución es marcadamente asimétrica. Cuando  $\sigma$  toma valores cercanos a la unidad la distribución se asemejará a la exponencial [Montgomery *et al.*, 2003], mientras que por el contrario si se aproxima a cero tenderá a parecerse más a una distribución normal.

La distribución Log-normal tiene especial interés para modelar comportamientos de variables que solo toman valores positivos, como es el tiempo. Varios estudios corroboran la similitud de la variabilidad de los tiempos de recorrido en entornos reales con la distribución Log-normal. En [Kaparias *et al.*, 2008] se asume esta distribución para evaluar el máximo retraso que puede alcanzarse en los recorridos y así mejorar la fiabilidad de su algoritmo de enrutamiento. Los trabajos de [De-feng *et al.*, 2011] y [Mahdi, 2011] demuestran que los tiempos de recorrido registrados experimentalmente obedecen a una distribución de este tipo. Sin embargo, en [Hoitner *et al.*, 2012] se registran los tiempos de recorrido de un determinado tramo de red vial en San Francisco y propone una distribución de probabilidad empírica que reproduce mejor los resultados experimentales que la distribución Log-normal.

Teniendo en cuenta lo anterior, la distribución Log-normal puede ser la más indicada para modelar la incertidumbre en los tiempos de recorrido de persecuidor y convoy en el problema objeto de estudio de esta tesis. Lo cual habría que validar, en un trabajo futuro, tras un conjunto suficiente de ensayos en condiciones de tráfico real.

Con todo, cabe plantear una primera aproximación del trabajo desarrollado en la tesis al caso de modelado de incertidumbre en los tiempos de recorrido de las unidades de transporte mediante funciones de distribución Log-normal.

Sean  $t_p$  y  $t_c$  los tiempos reales invertidos por la unidad persecuidora y por el convoy respectivamente para recorrer un tramo simple; tiempos que responden a variables aleatorias con función de distribución de probabilidad Log-normal. Mediante la siguiente transformación

$$\bar{t}_p = \ln(t_p), \quad \bar{t}_c = \ln(t_c) \quad (4.18)$$

se obtendrían unas nuevas variables temporales  $\bar{t}_p$  y  $\bar{t}_c$  con función de distribución normal y a las que sería de aplicación los algoritmos desarrollados en la tesis para calcular el nodo de enlace óptimo y la ruta más eficiente.

La validez de esta propuesta y el planteamiento de otras alternativas se incluyen en el conjunto de líneas de investigación futuras derivadas del trabajo aquí desarrollado.

#### 4.6.2. Función de distribución Gamma

La distribución Gamma también es empleada para modelar variables que presentan una mayor densidad de probabilidad a la izquierda que a la derecha de su valor medio y que ha demostrado ajustarse bien a la naturaleza de los tiempos de recorrido [Polus, 1979].

Una variable aleatoria  $x$  responde a una distribución Gamma si su función de densidad tiene la forma [Mood *et al.*, 1974]:

$$f(x) = \frac{\lambda}{\Gamma(\alpha)} (\lambda x)^{\alpha-1} e^{-\lambda x}, \quad \Gamma(\alpha) = \int_0^{\infty} e^{-x} x^{\alpha-1} dx, \quad 0 < x < \infty, \alpha > 0, \lambda > 0 \quad (4.19)$$

Siendo  $\Gamma(\alpha)$  la llamada función gamma,  $\alpha$  se denomina “parámetro de forma” de la distribución, que con valores muy cercanos a la unidad hace que la forma de la curva se acerque a la de la función de distribución exponencial (descrita en el apartado siguiente) mientras que para valores mayores adopta una forma similar la de la campana de Gauss. El segundo término  $\lambda$  se denomina “parámetro de escala” y define la asimetría de la curva, para valores elevados la distribución tendrá más densidad en el extremo derecho de la curva y para valores más pequeños logran una forma más simétrica de la misma. La media de la distribución está definida por  $\frac{\alpha}{\lambda}$  y su varianza por  $\frac{\alpha}{\lambda^2}$ .

Ejemplos de su aplicación en modelado de tiempos de recorrido se encuentran en [Nie *et al.*, 2012], [Hua *et al.*, 2010], [Jeong *et al.*, 2011]. Con todo, esta distribución es muy parecida a la

Log-normal, siendo en ocasiones difícil identificar cuando un conjunto de datos obedecen a una u otra distribución, problema que ya fue abordado por [Kundu *et al.*, 2005].

#### 4.6.3. Función de distribución Exponencial

Otra distribución también empleada para modelar tiempos de recorrido en diversos estudios es la Exponencial [Montgomery *et al.*, 2003]. Realmente es un caso particular de la distribución Gamma, cuando  $\alpha = 1$ . Su función de densidad tiene la forma:

$$f(x) = \lambda e^{-\lambda x}, 0 \leq x < \infty, \lambda > 0 \quad (4.20)$$

Y tiene media y varianza de  $\frac{1}{\lambda}$  y  $\frac{1}{\lambda^2}$ , respectivamente.

En la bibliografía se encuentran ejemplos de su aplicación en el ámbito del transporte. En [Talley *et al.*, 1987] se recurre a ella para estudiar la fiabilidad del servicio de autobuses, y obtener la probabilidad con que estos llegan a una parada determinada fuera de la hora programada, ya sea anticipándose o retrasando su llegada. En [Yang *et al.*, 2010] se justifica que el tiempo entre sucesivos vehículos y la distancia inter-vehículo obedecen a este tipo de distribución.



# 5

## DEMOSTRADOR CON ROBOTS PIONEER P3-DX

---

En este capítulo se describen los resultados de aplicar las estrategias presentadas a un caso de enrutamiento real. Para ello, se diseñó y construyó un demostrador de laboratorio en el que las unidades de transporte son robots móviles reales. Uno de los robots realiza funciones de líder del convoy y el otro asume las funciones de unidad perseguidora. El escenario de transporte se ha diseñado a escala y está adaptado a los prototipos robóticos. Los resultados obtenidos confirman la efectividad de la estrategia a nivel experimental y su posible extrapolación a un caso práctico.

### 5.1. Introducción

El capítulo anterior incluye algunos resultados de aplicar la estrategia de enrutamiento propuesta en un entorno virtual, diseñado y ejecutado mediante la plataforma de simulación Player/Stage. Para reforzar estos resultados y verificar la efectividad de la estrategia fuera del entorno de simulación, este capítulo se centra en probar los mismos algoritmos descritos en un entorno de trabajo real, empleando como unidades de transporte robots Pioneer P3-DX disponibles en el grupo de investigación GEINTRA [Geintra, 2012].

Para llevar a cabo esta demostración se diseñó un entorno de dimensiones 9x9m y formado por 21 nodos, 12 periféricos y 9 interiores. El escenario se ubicó en la zona interior del Edificio Politécnico de la Universidad de Alcalá y, aunque limitado, permite demostrar los aspectos más relevantes de la estrategia presentada en esta Memoria. La configuración del ensayo seguido es equivalente al presentado en las simulaciones del capítulo anterior; una unidad móvil situada en el interior del mapa debe enlazarse a un convoy que recorre su periferia y es representado por una única unidad móvil haciendo funciones de líder.

En la Figura 5.1 se muestra un diagrama en bloques de todos los elementos que intervienen en la demostración.



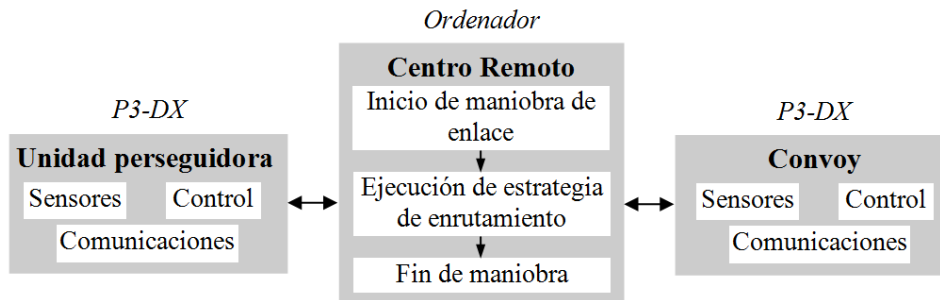


Figura 5.1. Diagrama con los componentes del demostrador.

## 5.2. Descripción del demostrador

En este apartado se detallan los aspectos más relevantes de los elementos del demostrador y del entorno de trabajo aprovechando los recursos robóticos del grupo de investigación. Durante la demostración se realizaron varias pruebas aplicando diferentes factores de riesgo, evaluando y contrastando los resultados con los obtenidos previamente a nivel de simulación en un entorno equivalente.

### 5.2.1. Elementos

A continuación se detallan los diferentes elementos que integraron el demostrador construido (Véase Figura 5.1), al que se ha añadido también la parte de comunicaciones entre las unidades y el Centro Remoto.

#### Centro Remoto

En el demostrador el Centro Remoto es un ordenador dotado con capacidad para comunicarse con las unidades móviles y realizar todas las operaciones de cálculo necesarias en la ejecución de la estrategia. Las tareas fundamentales llevadas a cabo por este siguen el esquema presentado en la Figura 4.8 del capítulo anterior, y son:

1. Ejecutar los algoritmos de enrutamiento: cálculo del nodo óptimo de enlace y la ruta eficiente.
2. Ejecutar algoritmos de cálculo y actualización de los parámetros estadísticos: tiempos medios y varianzas de cada tramo recorrido por cualquiera de las unidades móviles.
3. Almacenar toda la información relevante para la estrategia: características del mapa, experiencias temporales, tiempos medios y varianzas, etc.
4. Comunicarse periódicamente con todas las unidades que intervienen en la maniobra de enlace. Así, El CR tiene conocimiento de la ubicación de todas las unidades móviles, velocidades de recorrido, etc.

En la Figura 5.2 puede verse el CR junto al entorno en el que se ejecutó la demostración.



Figura 5.2. Centro Remoto y vista parcial del entorno diseñado para la demostración.

#### Unidades robóticas: Robots Pioneer P3-DX

El Pioneer P3-DX [P3-DX, 2012] de Mobile Robots es un robot móvil muy popular en tareas de investigación, debido a su versatilidad y por ser totalmente programable. Esto le permite abarcar un amplio abanico de aplicaciones, como por ejemplo, navegación autónoma, localización, control remoto, manipulación, mapeo, entre otras.

El robot (véase figura 5.3) está formado por un cuerpo de aluminio de aproximadamente 46x38x22cm, cuenta con dos ruedas activas de 16.5 mm con tracción diferencial alcanzando una velocidad lineal máxima de 1.2 m/s.

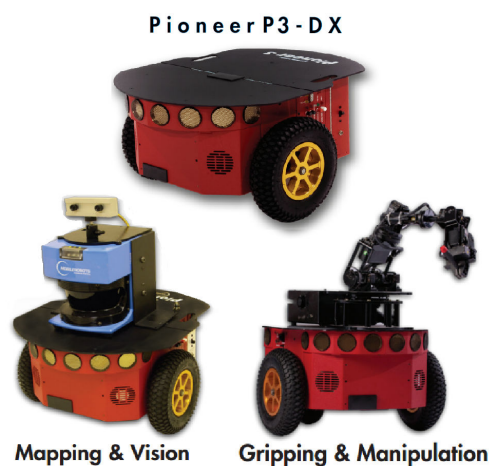


Figura 5.3. Plataforma robótica Pioneer P3-DX y algunas de sus posibles aplicaciones [P3-DX, 2012].

Su sistema sensorial básico está compuesto por un conjunto frontal de 8 sensores de ultrasonido con un ángulo de cobertura de hasta 180 grados. Además, cada rueda activa cuenta con un encóder de 500 ppv y un controlador *PID* de velocidad [Espinosa et al, 2011]. Por su parte, la batería permite una autonomía de unas 3 horas aproximadamente.

Información más detallada sobre la estructura de este robot puede encontrarse en [Salazar, 2011], [Santos, 2010], [Espinosa et al., 2010].

Además de los elementos básicos mencionados, para los diferentes trabajos de investigación llevados a cabo en el grupo GEINTRA, se fueron incorporados otros dispositivos y sensores:

1. Via-Epia (mini-ITX) con Linux empotrado [Epia, 2012] que realiza las funciones de computación asociadas al sensado, control de alto nivel y comunicaciones del robot.
2. Convertidor Ethernet Buffalo Wireless-G MIMO, WLI-TX4-G54HP-2, de hasta 125 Mbps [Buffalo, 2012], que permite al robot las funcionalidades de un nodo de comunicación inalámbrica con el CR.
3. Scanner Laser Hokuyo URG-04-LX [Hokuyo, 2012], en esta aplicación se le ha encomendado la detección de obstáculos y esquinas hasta alcanzar al convoy; y de mantener una distancia de seguridad con la unidad precedente una vez forma parte del mismo.
4. Disco duro para almacenamiento de programas y datos; y convertidor DC-DC para alimentar la electrónica incorporada a partir de las baterías del propio robot.
5. Sistema operativo de tiempo real Ubuntu kernel 2.6.23.

En la Figura 5.4. puede observarse la estructura de la unidad robótica original y los elementos añadidos hasta completar la versión de prototipo utilizada en los ensayos experimentales de esta Tesis.

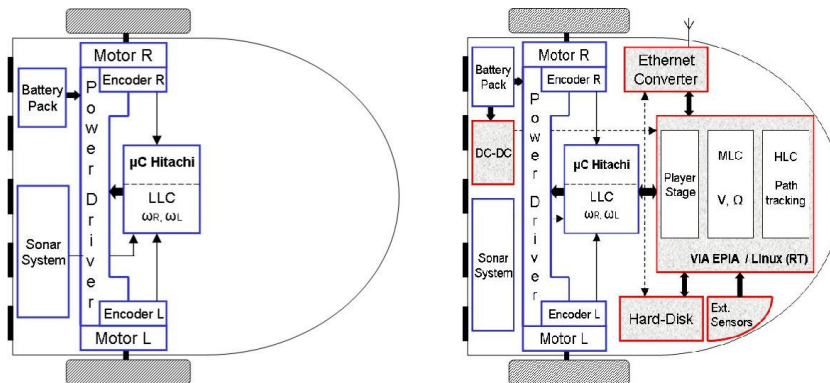


Figura 5.4. Componentes añadidos al Pioneer P3-DX para la realización de experimentos.

En la Figura 5.5 se muestra el aspecto real de ambas versiones, incluyendo vistas parciales de la arquitectura electrónica interna.

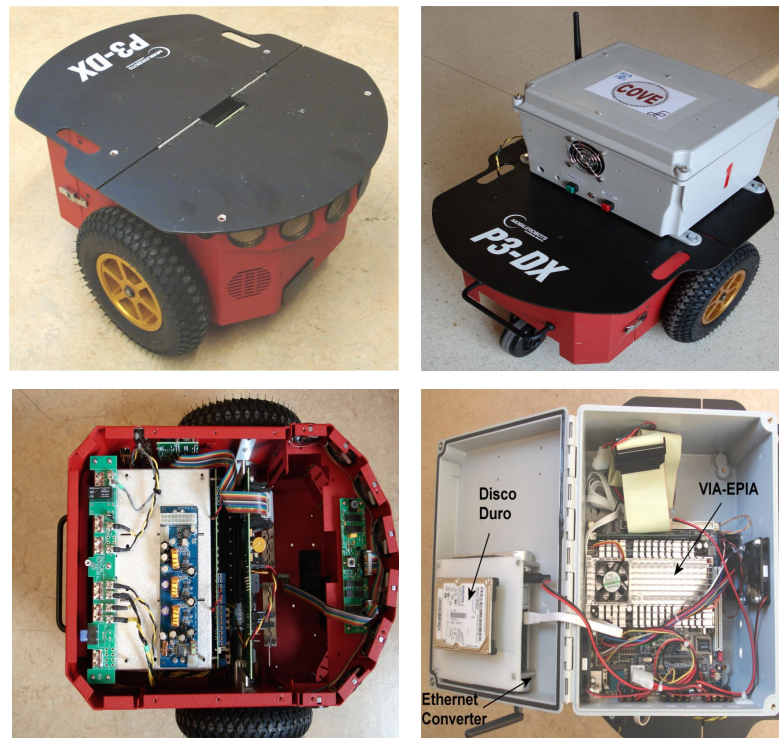


Figura 5.5. Plataforma básica (izda) y mejorada (dcha) en el grupo de investigación de robot P3-DX.

### Red de comunicaciones

Cada robot está equipado con una tarjeta Buffalo Wireless-G MIMO que permite comunicarse de forma inalámbrica con el CR (nodo maestro) de forma directa y con otros robots a través del nodo maestro. En la Figura 5.6 se muestra el esquema de comunicaciones aplicado.

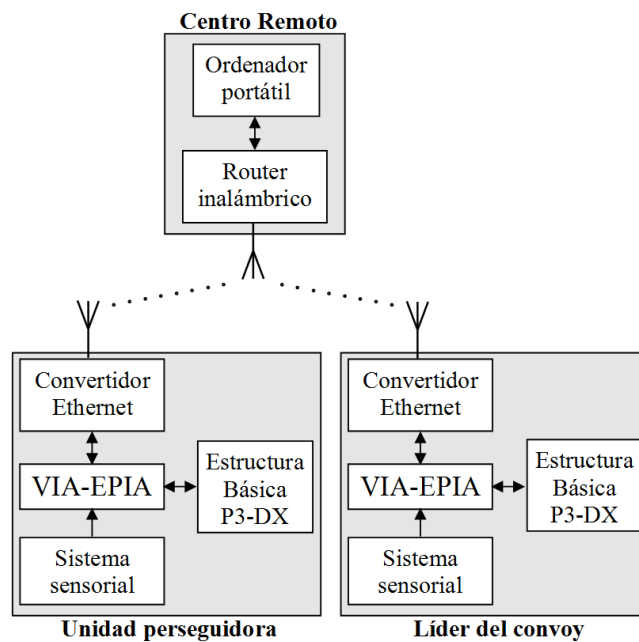


Figura 5.6. Arquitectura de comunicaciones de las unidades móviles con el CR.

Para la aplicación aquí descrita, la unidad perseguidora y el convoy no requieren comunicación entre sí, pero ambas intercambian información con el CR, que es quien gestiona toda la estrategia de enrutamiento.

### 5.2.2. Entorno

El entorno de trabajo para el experimento consistió en una zona de perímetro cuadrado de 9x9m. Las calles se delimitaron por estructuras cúbicas de un metro de lado, concretamente se utilizaron 16 elementos cúbicos organizados en 4 filas y 4 columnas, tal como muestra la Figura 5.7.

Dada las restricciones dimensionales del mapa planteado para las unidades robóticas P3-DX, la configuración de la Figura 5.7 es la que permite una mayor complejidad de enrutamiento. Así, se obtienen tramos entre nodo y nodo de dos metros de largo por uno de ancho.

En la Figura 5.7 puede verse una sección del entorno real utilizado (derecha) y una representación del mismo mostrando la distribución e identificación de nodos (izquierda). Para emular el efecto de la incertidumbre en los tiempos de recorrido, como se hizo a nivel de simulación en el capítulo anterior, el mapa fue dividido en tres zonas, atendiendo a los valores estadísticos de velocidad (media y varianza) de recorrido a aplicar a las unidades, como se indica en la parte izquierda de la Figura 5.7. Cada vez que un robot alcanza un nuevo nodo, el CR le asigna un valor aleatorio de velocidad de acuerdo con los parámetros estadísticos indicados.

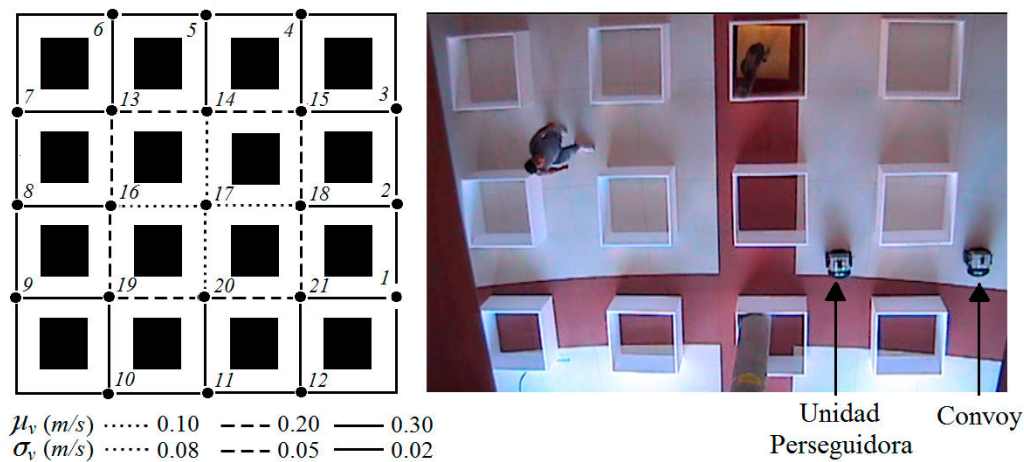


Figura 5.7. A la izquierda, representación gráfica del escenario del experimento, con las tres áreas claramente diferenciadas por el valor medio y varianza de la velocidad a la cual pueden moverse las unidades robóticas. A la derecha, una vista parcial del escenario real empleado. Uno de los robots hace las veces de unidad perseguidora mientras el otro asume el rol de líder del convoy.

### 5.3. Resultados experimentales

En este apartado se incluyen los resultados experimentales de dos pruebas realizadas con el demostrador ya comentado. Dada la limitada complejidad del escenario de transporte, para evaluar el efecto del factor de riesgo se ha optado por mostrar los dos casos extremos:

situación más conservadora ( $\delta=0$ ) y más arriesgada ( $\delta=1$ ), partiendo de idénticas condiciones iniciales.

### 5.3.1. Ensayo experimental con factor de riesgo del 0 %

La Figura 5.8 resume la primera demostración ( $\delta=0$ ) mediante una secuencia de instantáneas. En la parte derecha aparecen las vistas del escenario real en los instantes de tiempo seleccionados para destacar la evolución de la prueba. En el lado izquierdo puede observarse una representación del entorno de trabajo y la ubicación de las unidades que se mueven por este, en los instantes de tiempo seleccionados. La posición y orientación de las unidades de transporte se representan con triángulos: negro para el líder del convoy y blanco para la unidad perseguidora. Con un trazo continuo se indica el recorrido ya realizado por ambas unidades hasta el instante mostrado, y con un trazo discontinuo la trayectoria prevista por el algoritmo de enrutamiento para la unidad perseguidora hasta alcanzar el nodo óptimo de enlace calculado por el CR con los datos disponibles hasta ese momento. Obsérvese que el nodo de enlace puede cambiar de una instantánea a otra, en función del factor de riesgo y de cómo se ha desarrollado la prueba (velocidades asignadas aleatoriamente a las unidades de transporte).

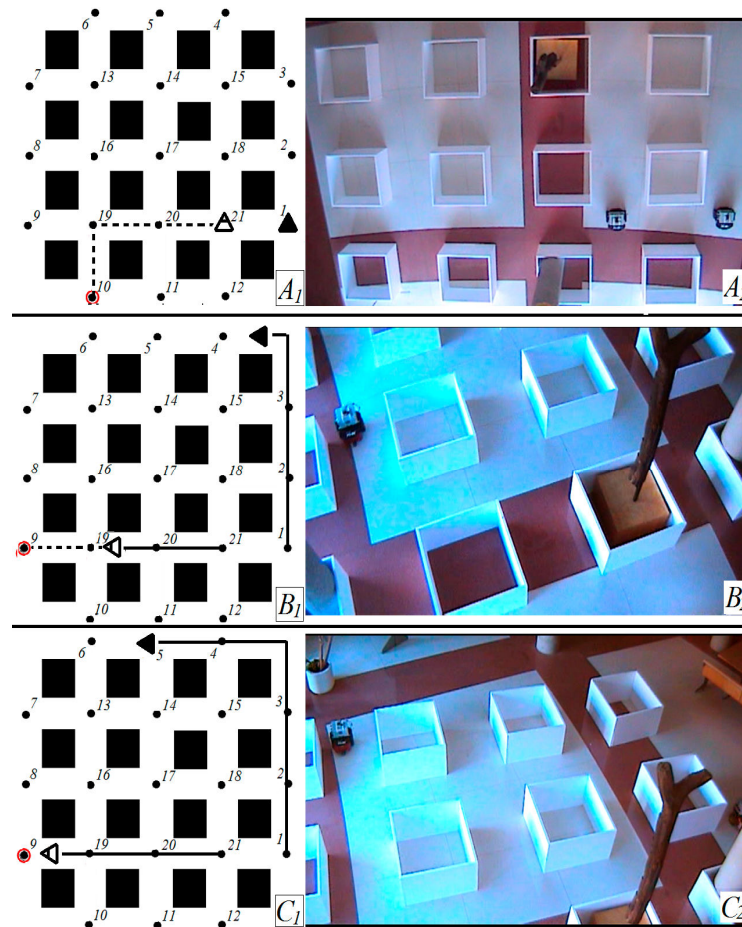


Figura 5.8. Demostración de la estrategia de enrutamiento implementada en robots Pioneer P3-DX, aplicando un Factor de Riesgo del 0 % ( $\delta = 0$ ). A la izquierda, las ubicaciones simultáneas de ambas

*unidades y el nodo de enlace calculado en cada momento. A la derecha, varias instantáneas tomadas durante el experimento, correspondientes a los mismos instantes.*

La Figura 5.8.A muestra la configuración inicial del experimento. El convoy está situado en el nodo 1, en dirección al nodo 2. La unidad perseguidora está localizada en el nodo 21, dirigida hacia el nodo 18. De acuerdo con esto, la unidad perseguidora determina que el mejor nodo de enlace es el número 10, y la mejor ruta para alcanzarlo es la marcada en línea discontinua en la Figura 5.8.A1.

La Figura 5.8.B muestra imágenes del experimento cuando la unidad perseguidora se aproxima al nodo 19. En ese instante el mejor nodo de enlace es re-evaluado, resultando  $s_e = 9$  y por consiguiente la trayectoria al mismo es re-planificada.

Finalmente, la Figura 5.8.C muestra el instante en el que la unidad perseguidora alcanza el nodo de enlace, y el convoy acaba de superar el nodo 5. En esta figura se puede apreciar claramente que el convoy se mueve más rápido que la unidad perseguidora. Esto se debe a que la distribución gaussiana que caracteriza las velocidades de la zona periférica tiene un valor medio mayor y menor varianza que las distribuciones gaussianas de las zonas interiores (véase Figura 5.7.izda).

De este primer experimento se puede concluir que el mejor nodo elegido para la maniobra de enlace es muy conservador, ya que está lejos de la posición inicial del convoy, así que la unidad perseguidora lo alcanza mucho antes que el convoy. La consecuencia directa de un bajo factor de riesgo es una baja probabilidad de fallo y un mayor tiempo de espera del perseguidor en el nodo de enlace para formar parte del convoy.

### **5.3.2. Ensayo experimental con factor de riesgo del 100 %**

A través de un segundo experimento (Figura 5.9) se compara el efecto de asumir un mayor riesgo de fallo en la maniobra de enlace. Las ubicaciones iniciales de los robots en este nuevo experimento son las mismas que en el anterior, excepto el valor del factor de riesgo que en este caso es  $\delta = 1$ .

Siguiendo con el mismo planteamiento, la Figura 5.9 contiene una secuencia de imágenes correspondientes a diferentes instantáneas tomadas durante el experimento: ubicación inicial de los robots, dos instantes intermedios y la situación final, cuando el perseguidor alcanza el nodo de enlace.

La Figura 5.9.A muestra el estado inicial. El mejor nodo de enlace (estimado por la unidad perseguidora) es el nodo 8, un nodo más cercano al convoy que el obtenido inicialmente en la demostración anterior (Figura 5.8.A).

En la Figura 5.9.B1 la unidad perseguidora ha girado 90 grados cuando aún se localiza en el nodo 21. Sin embargo al finalizar el giro, ha transcurrido un tiempo determinado y por tanto es necesario realizar una re-evaluación del nodo  $s_e$  inicialmente calculado, ya que el convoy ya ha recorrido cierta distancia. En este caso la unidad perseguidora considera que la mejor

opción para la maniobra de enlace es el nodo 9 (una solución más conservadora que la previamente estimada). Este cambio en el nodo de enlace supone la re-planificación de la mejor trayectoria para alcanzarlo (línea discontinua de la Figura 5.9.B1).

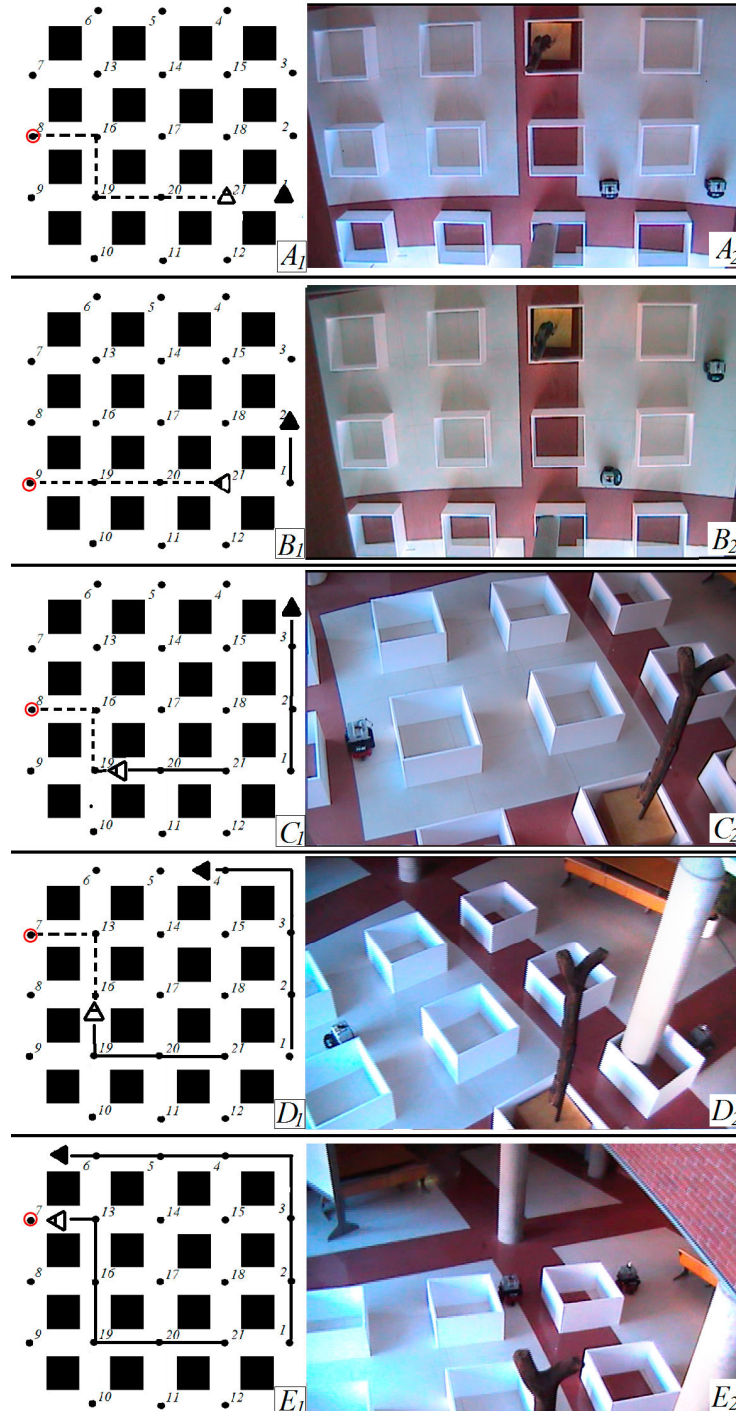


Figura 5.9. Resultados obtenidos por la estrategia de enrutamiento aplicando un Factor de Riesgo del 100 % ( $\delta = 1$ ). A la izquierda se representan las ubicaciones de ambas unidades robóticas a diferentes instantes de tiempo, así como el nodo de enlace elegido en cada momento. A la derecha pueden apreciarse instantáneas tomadas durante el experimento, correspondientes a los mismos instantes de tiempo.



La Figura 5.9.C, muestra la situación en la cual la unidad perseguidora está cerca de alcanzar el nodo 19. Al hacerlo se produce una nueva re-evaluación de objetivos, resultando nuevamente como nodo de enlace el nodo 8. Ya que esta solución es nuevamente diferente a la anterior, se debe re-planificar la ruta de enlace a seguir (línea discontinua en la Figura 5.9.C1).

En la Figura 5.9.D1, la unidad perseguidora está próxima al nodo 16, una vez más el nodo de enlace es re-evaluado considerando toda la información disponible, y es cambiado al nodo 7. Este nuevo valor de  $s_e$  está más cerca de la ubicación inicial y de la actual del convoy que en el cálculo anterior. Por otro lado, esta situación de objetivos (nodo de enlace y mejor ruta al mismo) son los definitivos del experimento.

Finalmente, la Figura 5.9.E muestra el instante de llegada de la unidad perseguidora al nodo de enlace final y la ubicación del convoy. Se puede ver claramente que la estrategia de enrutamiento propuesta tiende a penalizar los tramos centrales del mapa, pues en el mismo tiempo el recorrido del convoy es mayor.

El segundo experimento revela claramente que a mayor factor de riesgo, más rápidas son las maniobras de enlace, y menor será el tiempo que la unidad perseguidora deberá esperar al convoy en el nodo de enlace resultante (compárense las Figuras 5.8.C1 y 5.9.E1). Con todo, el número de veces que el nodo de enlace elegido cambia debido a las re-evaluaciones se incrementa cuando el factor de riesgo aumenta, y por consiguiente también es mayor el número de re-planificaciones de la ruta de enlace.

Finalmente, es importante mencionar que el experimento mostrado en la Figura 5.9 es un ejemplo de maniobra de enlace exitosa. Sin embargo debemos tener presente el hecho de que a mayor factor de riesgo, mayor es la probabilidad de que la unidad perseguidora llegue tarde al nodo de enlace calculado resultando en una maniobra fallida.

Los experimentos completos, de los que se han obtenido las instantáneas de las Figuras 8 y 9 están disponibles en la web del Grupo Geintra de la Universidad de Alcalá [Geintra, 2012].

# 6

## CONCLUSIONES Y TRABAJOS FUTUROS

---

Este capítulo recoge las conclusiones más relevantes extraídas de esta investigación. Además, se apuntan las principales líneas futuras de trabajos derivadas de la misma.

### 6.1. Conclusiones

El desarrollo de esta tesis ha permitido aportar soluciones a la maniobrabilidad de vehículos en entornos urbanos. Especialmente, se aborda el caso de unidades que una vez realizada su tarea de forma autónoma e independiente han de incorporarse a un convoy circulando continuamente por una trayectoria que circunda el escenario de transporte. Para que la maniobra de enlace se realice con éxito se han de superar dos importantes retos: elegir el nodo óptimo de enlace y calcular la ruta más eficiente hasta el mismo, todo ello considerando la incertidumbre asociada a los tiempos de recorrido entre nodos internos y periféricos. La maniobra se realiza con éxito cuando el perseguidor llega al nodo de enlace antes que el convoy y en un tiempo mínimo. Situación que no se puede garantizar en todos los casos debido a la incertidumbre inherente al problema. Para modular la probabilidad de éxito se introduce un término denominado Factor de Riesgo, estableciendo una relación inversa entre probabilidad de fallo y tiempo de ejecución de la maniobra.

La estrategia propuesta, basada en Programación Dinámica, se caracteriza por su adaptación a entornos cambiantes re-planificando objetivos cuando se requiere, y robustez frente a incertidumbres en los tiempos de recorrido.

La solución descrita en la tesis ha generado las siguientes contribuciones científicas y tecnológicas:

1. **Definición de la propuesta y planificación de la solución.** La alternativa al tráfico convencional en entornos especiales defendida en la tesis consiste en un convoy de unidades circulando por la periferia del escenario de transporte. Las unidades que lo forman se separan del mismo para acceder a un punto en el interior del mapa según la misión encomendada. A partir de ahí, la unidad ha de incorporarse, de forma automática al convoy circulando por la trayectoria periférica.

Teniendo en cuenta los condicionantes de problema se plantea una solución administrada por un centro remoto. Este se encarga de almacenar la topología del entorno, registrar las experiencias de los tiempos de recorrido de las diferentes unidades de transporte, actualizar los parámetros estadísticos asociados y ejecutar la algoritmia cuando es requerida por la unidad que decide iniciar la maniobra de enlace. Todas las unidades del escenario de transporte mantienen una comunicación inalámbrica con el centro remoto.

Las publicaciones relacionadas que validan esta contribución son: [Valdés *et al.*, 2009] y [Valdés *et al.*, 2006].

2. **Cálculo el nodo óptimo de enlace y ruta eficiente hasta el mismo.** Para el nodo periférico óptimo de enlace se calcula el tiempo estimado que unidad perseguidora y convoy tardan en alcanzar cada uno de los nodos periféricos y se elige como solución aquél que reduce el tiempo de enlace de la unidad perseguidora con el convoy.

El algoritmo que proporciona la ruta de enlace desde la ubicación actual de la unidad perseguidora va eligiendo, de todos los nodos sucesores posibles, solo aquél cuyo tiempo estimado al nodo periférico ya elegido es mínimo. El método de búsqueda ofrece dos ventajas principales: utiliza el mínimo número de nodos explorados y, por tanto, se reducen los requisitos electrónicos y computacionales necesarios para su ejecución en tiempo real, de especial interés en el caso de mapas complejos.

Para el diseño de ambos algoritmos que aplican técnicas recursivas de Programación Dinámica, lo que permite calcular y actualizar rápidamente la estimación de los tiempos de recorrido entre cada par de nodos cualesquiera que componen el mapa, a partir de los tiempos reales de recorrido de tramos simples (entre nodos consecutivos) aportados por las unidades del entorno.

Las publicaciones relativas a esta contribución son: [Valdés *et al.*, 2011b] y [Valdés *et al.*, 2011c].

3. **Modelado de incertidumbre en los tiempos de recorrido.** En este trabajo se ha recurrido a funciones de distribución gaussianas para modelar la incertidumbre en los tiempos de recorrido propia de un escenario de transporte real. De esta forma se facilita la propagación de valores estadísticos temporales, medias y varianzas, a nodos no adyacentes internos y externos del mapa. Por otra parte, la información de tiempos de recorrido proporcionada por todas las unidades del escenario de transporte contribuye actualizar dinámicamente los citados parámetros estadísticos. Actualización que puede realizarse de forma off-line para no cargar el canal de comunicación con el centro remoto y aprovechar momentos de menor carga computacional de este.

La consideración de esta incertidumbre ha obligado a que cada vez que la unidad perseguidora alcanza un nuevo nodo, se re-evalúe el nodo óptimo de enlace teniendo en cuenta las nuevas ubicaciones de perseguidor y convoy. Si cambia la elección del nodo de enlace se ha de proceder a la re-planificación de la trayectoria hasta el mismo. Estas propiedades de robustez y adaptación son claras contribuciones de la solución propuesta.

En relación con esta contribución cabe destacar las siguientes publicaciones: [Valdés *et al.*, 2011c] y [Valdés *et al.*, 2012a].

4. **Factor de Riesgo.** Dada la imposibilidad de garantizar con toda seguridad el éxito de la maniobra teniendo en cuenta la naturaleza estadística del problema planteado, en la tesis se introduce un factor que permite limitar el riesgo de fallo en la ejecución de la maniobra de enlace. En la medida en que el Factor de Riesgo se aproxima el mínimo (0 %) el algoritmo proporciona nodos de enlace más conservadores, lo que se traduce en un menor número de re-planificaciones, tiempos mayores para realizar la maniobra y menor probabilidad de fallo.

Las publicaciones relacionadas con esta contribución son: [Valdés *et al.*, 2012a] y [Valdés *et al.*, 2012b].

5. **Demostrador de la estrategia de enrutamiento.** Todos los algoritmos que integran la estrategia han sido validados a nivel de simulación utilizando la plataforma Player&Stage, primero manteniendo un tiempo de recorrido conocido entre nodos y después aplicando una distribución gaussiana a dicho tiempo. Superada esta etapa se ha diseñado un demostrador de los algoritmos con un escenario de transporte en el que las unidades móviles son robots Pioneer P3-DX.

Las siguientes publicaciones recogen diferentes aspectos relacionados con las pruebas experimentales realizadas para la consecución de los resultados mostrados en la tesis: [Espinosa *et al.*, 2008a], [Espinosa *et al.*, 2008b], [Valdés *et al.*, 2008], [Espinosa *et al.*, 2010], [Espinosa *et al.*, 2011].

## 6.2. Líneas futuras de investigación

La investigación en las líneas resumidas en el punto anterior ha permitido abrir nuevas vías relacionadas de exploración científica que se concretan en este capítulo y que podrá servir como punto de partida para nuevos trabajos de investigación.

- **Estrategia de enrutamiento para maniobra de separación del convoy.** Se trata de la maniobra complementaria a la propuesta en esta tesis, en la cual uno

de los vehículos seguidores que integran el convoy debe abandonarlo para dirigirse a un nodo interior específico. Se requiere una estrategia para obtener el nodo periférico óptimo a partir del cual esta unidad pueda abandonar el convoy y dirigirse a la ubicación deseada, a través de la ruta óptima, en función del criterio de optimalidad elegido, ya sea tiempo de recorrido, distancia mínima u otro.

- **Incorporación de aprendizaje en la elección del Factor de Riesgo.** Hasta ahora, el Factor de Riesgo aplicado ha sido elegido por el diseñador y se ha mantenido invariante durante la realización de la maniobra de enlace. Sin embargo, nos parece interesante explorar la posibilidad de desarrollar alguna estrategia que permita no solo automatizar la elección del valor del Factor de Riesgo, sino también lograr que este varíe de forma dinámica conforme la unidad perseguidora recorre la ruta hacia la vía periférica por la que se mueve el convoy. Es razonable pensar que el factor de riesgo podría ser elevado cuando la unidad perseguidora todavía se encuentra lejos de la vía periférica (la existencia de muchos nodos intermedios facilita la re-planificación y minimiza la probabilidad de fallo), mientras que debería reducirse a medida que la unidad perseguidora se aproxima al convoy. En este último caso el factor de riesgo debería ser bajo generándose así soluciones más conservadoras.

Uno de los paradigmas más adecuados para este propósito sería el del aprendizaje por refuerzo [Sutton *et al.*, 1998]. En nuestro caso el refuerzo se obtendría como resultado de comparar el nodo en el que se produce la maniobra de enlace, con el nodo en el que dicho enlace se produciría si el factor de riesgo elegido fuese del 0 %. Obviamente el hecho de que la maniobra haya o no haya tenido éxito también condicionará el valor del refuerzo. Si bien cualquiera de los algoritmos de aprendizaje es válido para nuestro propósito (*TTD*( $\lambda, m$ ) [Cichosz, 1997], *Q-learning* [Watkins, 1989],...), nuestra primera intención será probar el uso de los algoritmos descritos en [Quintía *et al.*, 2010], dado que permitirían el aprendizaje simultáneo del espacio de estados y el valor del factor de riesgo que se debe tomar en cada uno de dichos estados.

Esta línea de incorporación de aprendizaje a la toma de decisiones de unidades en convoy ya ha sido iniciada por el grupo de trabajo del que forma parte el autor, como se refleja en la publicación [Rodríguez *et al.*, 2009].

- **Extender la maniobra de enlace al caso de más de una unidad perseguidora y más de un convoy.** En este caso se trata de la coexistencia de varias unidades perseguidoras que ejecutan la misma maniobra y/o varios convoyes en movimiento por la periferia, con lo cual se incrementa, por una parte el número de variables que intervienen en la toma de decisiones, y por otra la necesidad de establecer estrategias que eviten la obstaculización de recorridos, prioricen salida de cruces, etc. En este caso cada unidad perseguidora deja de formar parte de un único convoy y, en cada caso, se ha de tomar la decisión del convoy más apropiado

al que ha de enlazarse para reducir el tiempo de maniobra, salvo restricciones adicionales, como el número máximo de unidades por convoy.

- **Demostración de los algoritmos de enrutamiento con vehículos reales.** Hasta ahora, para validar los algoritmos defendidos en la tesis, se han utilizado robots P3-DX como prototipo de unidades de transporte. Sin embargo la adaptación de los algoritmos a vehículos reales supone superar dos etapas, una primera en simulación contando con modelos (cinemáticos y dinámicos) más acordes al comportamiento de vehículos convencionales, y por otra disponer de estos vehículos para las pruebas de campo.
- **Analizar el impacto de otros modelos de incertidumbre.** En el capítulo 4 se indica que existen en la literatura otras propuestas de modelado de incertidumbre en los tiempos de recorrido de vehículos en entornos urbanos mediante distribuciones no gaussianas (log-normal, gamma, exponencial, etc.). Se propone como nueva línea de investigación la evaluación del impacto de estas alternativas en los demostradores ya utilizados en esta tesis, máxime teniendo en cuenta que se pueden requerir varias re-planificaciones a lo largo del recorrido realizado por la unidad perseguidora.



## ANEXO 1

---

El siguiente análisis tiene como fin demostrar la eficiencia computacional del algoritmo propuesto, centrada en minimizar el tiempo de ejecución de la búsqueda de la ruta y los recursos de memoria implicados.

El objetivo de esta sección es demostrar que cuando un nodo evaluado es elegido como parte de la solución no será descartado en el futuro, y todos aquellos nodos que no son elegidos pueden desecharse definitivamente. En otras palabras, solo serán expandidos, y almacenados en memoria, aquellos nodos que formen parte de la ruta final devuelta por el algoritmo.

Para estudiar la eficiencia computacional se analiza la función de coste empleada en la selección de cada nodo, y que tiene la forma:

$$f(n_i) = T(n^0, n_i) + T(n_i, s_e) \quad (\text{A.1})$$

Está claro que al evaluar el nodo  $n_i$ ,  $T(n_i, s_e)$  representa el coste de alcanzar  $s_e$  desde  $n_i$ , y que equivale al término heurístico  $h(n)$  ya presentado en la sección 3.2.3.2. Recordando el concepto de admisibilidad, para que el algoritmo propuesto devuelva soluciones óptimas se debe cumplir que en cada nodo  $n_i$  evaluado:

$$T(n_i, s_e) \leq C(n_i, s_e) \quad (\text{A.2})$$

Donde  $C(n_i, s_e)$  representa el coste real de alcanzar al objetivo  $s_e$  desde  $n_i$ .

Relacionado con la admisibilidad está el concepto de consistencia de las funciones heurísticas. La condición de consistencia [Nilsson, 1998], [Pearl, 1984] significa que el coste de los nodos es monótonamente no-decreciente mientras la búsqueda avanza a lo largo de la ruta, partiendo del nodo inicial. Formalmente se puede expresar:

$$T(n_i, s_e) \leq C(n_i, n^h) + T(n^h, s_e), n^h \in \text{sucesores}(n_i) \quad (\text{A.3})$$

Siendo  $C(n_i, n^h)$  el coste real de ir desde  $n_i$  a su sucesor  $n^h$ .

Al emplear una heurística consistente, se asegura que cuando se expande un nodo  $n_i$  y se selecciona uno de sus sucesores  $n^h$ , se considera que se ha encontrado una solución óptima hasta dicho nodo [Russell et al., 2003] y por tanto no existirá una ruta más corta que la encontrada hasta ahora, que parte de  $n_0$ , y pasa por  $n_i$  y  $n^h$ .



Se define entonces  $SOL = \{sol(0), sol(1), sol(2), \dots, sol(L)\}$ , como el conjunto de nodos adyacentes que constituyen una solución  $SOL$  lograda aplicando la estrategia de enrutamiento propuesta, siendo  $L$  la profundidad de la búsqueda a la que se encuentra esta solución. De esta forma, la solución inicial,  $sol(0) = n^0$  coincide con el nodo inicial de la unidad perseguidora que será el primer elemento en la trayectoria que se obtenga, mientras  $sol(L) = s_e$  coincide con el nodo de enlace, que será el último. Consecuentemente,  $sol(i) \in \{n_1, \dots, n_M\}, \forall i = 1, \dots, L-1$  estará comprendida por nodos interiores. Finalmente, es cierto afirmar que  $sol(i+1) \in sucesores(sol(i)), \forall i = 1, \dots, L-1$ .

### Teorema P

$$\begin{aligned} f(sol(i)) &= f(sol(i-1)), \forall i = 1, \dots, L; \\ f(sol(i)) &\leq f(n_j), \\ \forall i = 1, \dots, L, \forall n_j &\in \{sucesores(sol(0)), sucesores(sol(1)), \dots, sucesores(sol(i-1))\} \end{aligned} \quad (A.4)$$

Para probar que cada solución obtenida por la estrategia propuesta verifica el teorema **P**, se va a emplear inducción matemática. Esto significa que es necesario demostrar que **P** es válido para el más bajo valor de  $i$  (caso base) y además se tiene que probar que si la solución del algoritmo a la profundidad  $i$  ( $sol(i)$ ) verifica **P**, entonces  $sol(i+1)$  también lo hace,  $\forall i \leq L-1$  (paso inductivo).

### Prueba del caso base

Recordando que  $n^0$  es la ubicación de la unidad perseguidora en el momento de ejecutar el algoritmo de búsqueda, de acuerdo a (Ec. 3.3) se puede obtener el coste  $f$  correspondiente de  $n^0$ :

$$f(n^0) = T(n^0, n^0) + T(n^0, s_e) = T(n^0, s_e) \quad (A.5)$$

Donde  $T(n^0, s_e)$ , tomando por referencia la ecuación recursiva de la Figura 3.8, se calcula como:

$$T(n^0, s_e) = \min \{T(n^0, n_i) + T(n_i, s_e)\}, \forall n_i \in sucesores(n^0) \quad (A.6)$$

Obviamente, ya que  $n^0$  es el único nodo que ha sido evaluado hasta el momento, será elegido como parte de la solución obtenida hasta entonces, de modo que:

$$Solución = \{sol(0) = n^0\} \quad (A.7)$$

En consecuencia, a continuación se deben generar los nodos sucesores de  $n^0$  para ser analizados según el coste asociado a cada uno, determinado mediante (Ec. 3.3):

$$f(n_i \in sucesores(n^0)) = T(n^0, n_i) + T(n_i, s_e) \quad (A.8)$$

Comparando (Ec. A.6) y (Ec. A.8) se puede concluir que existe al menos un nodo interior  $n^*$ , descendiente de  $n^0$ , que minimiza (Ec. A.6) y, por consiguiente, verifica (Ec. A.9):

$$\begin{aligned} a) f(n^*) &= T(n^0, s_e) = f(n^0), n^* \in \text{sucesores}(n^0) \\ b) f(n^*) &\leq f(n_i \in \text{sucesores}(n^0)) \end{aligned} \quad (\text{A.9})$$

Ya que  $n^*$  es el nodo con el coste mínimo está claro que el algoritmo de búsqueda lo elegirá para ser explorado (y por tanto sus descendientes serán evaluados). Así,  $n^*$  se convierte en parte de la solución alcanzada hasta el momento por lo que:  $\text{sol}(1) = n^* \in \text{sucesores}(n^0)$  y los costes de los sucesores de  $n^*$  serán calculados.

Es importante darse cuenta de que la (Ec. A.9) es precisamente la proposición **P** aplicada a  $\text{sol}(1)$ . Por tanto, se prueba que  $\text{sol}(1)$  verifica **P**.

### Prueba del paso inductivo

Ahora es necesario probar que si la  $\text{sol}(i)$  verifica **P**, entonces la  $\text{sol}(i+1)$  también lo hará. El hecho de que  $\text{sol}(i)$  verifica **P** significa:

$$\begin{aligned} a) f(\text{sol}(i)) &= f(\text{sol}(i-1)) = f(n^0) \\ b) f(\text{sol}(i)) &\leq f(n_i), n_i \in \{\text{sucesores}(\text{sol}(0)), \text{sucesores}(\text{sol}(1)), \dots, \text{sucesores}(\text{sol}(i-1))\} \end{aligned} \quad (\text{A.10})$$

Primeramente, se considera el coste en el nodo correspondiente a  $\text{sol}(i)$ :

$$f(\text{sol}(i)) = T(n^0, \text{sol}(i)) + T(\text{sol}(i), s_e) \quad (\text{A.11})$$

donde:

$$T(\text{sol}(i), s_e) = \min\{T(\text{sol}(i), n_i) + T(n_i, s_e)\}, \forall n_i \in \text{sucesores}(\text{sol}(i)) \quad (\text{A.12})$$

El coste asociado a cada nodo conectado al de la  $\text{sol}(i)$  es directamente determinado utilizando (Ec. 3.3):

$$f(n_i \in \text{sucesores}(\text{sol}(i))) = T(n^0, n_i) + T(n_i, s_e) \quad (\text{A.13})$$

Sin embargo, si se considera que la solución total hasta ahora consiste de:

$\text{sol}(1), \text{sol}(2), \dots, \text{sol}(i)$ , se puede re-escribir (Ec. A.13) como:

$$f(n_i \in \text{sucesores}(\text{sol}(i))) = T(n^0, \text{sol}(i)) + T(\text{sol}(i), n_i) + T(n_i, s_e) \quad (\text{A.14})$$

Comparando (Ec. A.13) y (Ec. A.14) se puede concluir fácilmente que existe al menos un nodo interior  $n'$ , sucesor del nodo en  $\text{sol}(i)$ , que minimiza (Ec. A.12) y por consiguiente verifica la siguiente afirmación:

$$\begin{aligned} a) f(n') &= f(\text{sol}(i)), n' \in \text{sucesores}(\text{sol}(i)) \\ b) f(n') &\leq f(n_i), \forall n_i \in \text{sucesores}(\text{sol}(i)) \end{aligned} \quad (\text{A.15})$$

Debido a que  $n'$  tiene el mismo coste que el nodo en  $\text{sol}(i)$  (Ec. A.15a), y  $\text{sol}(i)$  satisface **P** (Ec. A.10), está claro que:

$$\begin{aligned} f(n') &= f(\text{sol}(i)) = f(\text{sol}(i-1)) = \dots = f(n^0) \\ f(n') &\leq f(n_i \in \{\text{sucesores}(\text{sol}(0)), \text{sucesores}(\text{sol}(1)), \dots, \text{sucesores}(\text{sol}(i-1))\}) \end{aligned} \quad (\text{A.16})$$

De acuerdo con la afirmación anterior en (Ec. A.16), es correcto decir que  $n'$  se convierte en el próximo nodo de la solución alcanzada hasta el momento:  $sol(i+1) = n'$ , y todos los sucesores de  $n'$  serán elegidos como los próximos nodos a ser explorados. Esto prueba que **P** es también cierto para  $sol(i+1)$ .

Una conclusión muy importante puede ser inferida de la prueba del Teorema P: todos los nodos pertenecientes a la solución tendrán el mismo coste. De esta forma, cada vez que el algoritmo deba elegir el siguiente nodo a analizar de entre varios sucesores, seleccionará aquel que tenga el mismo valor de coste que el resto de nodos de la solución alcanzada hasta ese momento, y por tanto dicho nodo también pasará a formar parte de la solución óptima, mientras que el resto de nodos sucesores pueden ser descartados definitivamente. Evidentemente, esto sin duda favorece una reducción considerable del tiempo de ejecución del algoritmo y la cantidad de memoria requerida para almacenar los nodos analizados.

## BIBLIOGRAFÍA

---

- [Abbas *et al.*, 2010] M. K. Abbas, A. B. Abdullah. *Vehicles Speed Control via VANET*, In: ICMU2009, K.L, Malaysia. 2010.
- [Alazab *et al.*, 2011] A. Alazab, S. Venkatraman, J. Abawajy, M. Alazab. *An Optimal Transportation Routing Approach using GIS-based Dynamic Traffic Flows*. 3rd International Conference on Information and Financial Engineering IPEDR Vol. 12. IACSIT Press, Singapore. 2011.
- [Alton *et al.*, 2008] K. Alton, I. M. Mitchell. *Efficient Dynamic Programming for Optimal Multi-Location Robot Rendezvous*. Proceedings of the 47th IEEE Conference on Decision and Control. Cancun, Mexico, Dec. 9-11, 2008.
- [Ashtankar *et al.*, 2009] P. Ashtankar, S. Dorle, M. Chakole, D. Avinash, G. Keskar. *Design Approach for Anti-collision Mechanism between Vehicle to Vehicle for Improving Safety Operation In Intelligent Transportation System*. Proceedings of the International MultiConference of Engineers and Computer Scientists 2009. Vol. II.IMECS 2009, Hong Kong. March 18 - 20, 2009,
- [Azi *et al.*, 2007] N. Azi, M. Gendreau, J. Y. Potvin. *An exact algorithm for a single-vehicle routing problem with time windows and multiple routes*. European Journal of Operational Research, 178 (2007), pp: 755–766.
- [Bang-Jensen *et al.*, 2007] J. Bang-Jensen, G. Gutin. *Digraphs Theory, Algorithms and Applications*. Springer-Verlag. ISBN: 978-1-84800-997-4. 2007.
- [Bartin *et al.*, 2010] B. Bartin, K. Ozbay. *Determining the Optimal Configuration of Highway Routes for Real-Time Traffic Information: A Case Study*. IEEE Transactions on Intelligent Transportation Systems. Vol. 11, No. 1, Doi: 10.1109/TITS.2009.2029241. March 2010
- [Bellman, 1954] R. Bellman. *The theory of dynamic programming*. Bulletin of the American Mathematical Society 60. pp: 503–516, doi:10.1090/S0002-9904-1954-09848-8. 1954.
- [Bellman, 1957] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [Bellman, 1958] R. Bellman. *On a routing problem*. Quarterly Applied Mathematics, XVI(1). pp: 87–90. 1958.

- [Berry *et al.*, 1951] D. S. Berry, D. M. Belmont. *Distribution of vehicle speeds and travel times*. Proceedings of 2nd Berkeley symposium on mathematical and statistical probability, pp: 589-602. 1951.
- [Bond *et al.*, 2010] D. M. Bond, N. A. Widger, W. Ruml and X. Sun. *Real-Time Search in Dynamic Worlds*. Proceedings of the Third Annual Symposium on Combinatorial Search, 2010.
- [Box *et al.*, 1958] G. E. P. Box, M. E. Müller. *A Note on the generation of random normal deviates*. The Annals of Mathematical Statistics, Vol. 29, No. 2, pp: 610-611. Doi: 10.1214/aoms/1177706645. 1958.
- [Buffalo, 2012] WLI-TX4-G54HP. Datasheet available:  
[http://www.buffalotech.com/files/downloads/WLI-TX4-G54HP\\_DS.pdf](http://www.buffalotech.com/files/downloads/WLI-TX4-G54HP_DS.pdf)
- [Bulitko *et al.*, 2006] V. Bulitko, G. Lee. *Learning in Real-Time Search: A Unifying Framework*. Journal of Artificial Intelligence Research 25 (2006). pp: 119-157.
- [Bulitko *et al.*, 2008] V. Bulitko, M. Lustrek, J. Schaeffer, Y. Bjornsson, S. Sigmundarson. *Dynamic Control in Real-Time Heuristic Search*. Journal of Artificial Intelligence Research 32: 419 – 452. 2008.
- [Cai *et al.*, 2011] L. Y. Cai, D. Sha. *Shortest paths subject to time-varying stochastic transit times*. 8th International Conference on Service Systems and Service Management (ICSSSM). June 2011. pp: 1 - 2. ISSN : 2161-1890. Print ISBN: 978-1-61284-310-0.
- [California PATH, 2011] California Partners for Advanced Transit and Highways.  
<http://www.path.berkeley.edu/>
- [Cantone *et al.*, 2009] D. Cantone, S. Faro. *A Faster Algorithm for the Single Source Shortest Path Problem in the Presence of Few Sources or Destinations of Negative Arcs*. In Proceedings of the 35th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2009). Mlyn, Czech Republic. January 24-30, 2009.
- [Cazenave, 2006] T. Cazenave. *Optimizations of data structures, heuristics and algorithms for path-finding on maps*. IEEE Symposium on Computational Intelligence and Games, 22-24 May 2006. pp: 27 - 33. Reno, NV. Print ISBN: 1-4244-0464-9.
- [Chao, 2010] Y. Chao. *A developed Dijkstra algorithm and simulation of urban path search*. 5th International Conference on Computer Science and Education (ICCSE), 2010. Issue Date: 24-27 Aug. 2010. pp: 1164 - 1167. Hefei. Print ISBN: 978-1-4244-6002-1.
- [Cichosz, 1997] P. Cichosz. *Reinforcement learning by truncating temporal differences*. Ph.D. Thesis, Dpt. Of Electronics and Information Technology, Warsaw University of Technology, 1997.
- [COOPERS Project, 2011] Co-operative Systems for Intelligent Road Safety.  
<http://www.coopers-ip.eu/>

- [Cormen *et al.*, 2009] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms*, Third Edition. The MIT Press. 2009. ISBN-10:0-262-03384-4.
- [COVE, 2009] COVE Project. <http://www.geintra-uah.org/projects/sistema-transporte-inteligente-guiado-cooperativo-vehiculos-electricos-entornos-especiales>
- [CoVel Project, 2011] Cooperative Vehicle Localization for Efficient Urban Mobility. <http://www.covel-project.eu/>
- [CVIS Project, 2011] Cooperative Vehicle - Infrastructure Systems. <http://www.cvisproject.org/>
- [De Santiago *et al.*, 2010] L. De Santiago, F. Espinosa, M. A. Ruiz, J.A. Jiménez, E. Santiso, D. Sanguino, A. Wefky, W. G. Fano. *Effect of electrical vehicle-driver interaction on the radiated electromagnetic emissions: measurement methodology*. Int. Conf. on Industrial Technology IEEE-ICIT 2010 IEEE Catalog Number: CFP10CIT-USB. ISBN: 978-1-4244-5697-0. Doi 10.1109/ICIT.2010.5472579. pp: 1113-1118. Valparaiso, Chile. 2010.
- [De-feng *et al.*, 2011] L. V. De-feng, G. Xiu-cheng. *Study on Path Travel Time Reliability Model*. 2011 Fourth International Conference on Intelligent Computation Technology and Automation. Doi: 10.1109/ICICTA.2011.238. 2011.
- [Dell'Amico *et al.*, 2008] M. Dell'Amico, M. Iori, D. Pretolani. *Shortest paths in piecewise continuous time-dependent networks*. Operations Research Letters Vol. 36, Issue 6, pp: 688-691. November 2008.
- [Delling *et al.*, 2009] D. Delling, P. Sanders, D. Schultes, D. Wagner. *Engineering route planning algorithms*. In: Algorithmics of large and complex networks, Vol. 5515 of Lecture Notes in Computer Science, pp: 117-139. Springer-Verlag 2009.
- [Dijkstra, 1959] E. W. Dijkstra. *A note on two problems in connections with graphs*. Numerische Mathematik, 1:269-271, 1959.
- [Ding *et al.*, 2008] B. Ding, J. Xu Yu, L. Qin. Finding time-dependent shortest paths over large graphs. Proceedings of the 11th international conference on Extending database technology: Advances in database technology, March 25-29, 2008, Nantes, France. Doi: 10.1145/1353343.1353371.
- [Dreyfus, 1969] S. Dreyfus. *An appraisal of some shortest-path algorithms*. Operations Research, Vol. 17, No. 3 (May - Jun., 1969), pp: 395-412.
- [Epia, 2012] VIA EPIA Mini-ITX. <http://www.via.com.tw/en/initiatives/spearhead/mini-itx/>
- [Espinosa *et al.*, 2007] F. Espinosa, A.M.H. Awawdeh, M. Mazo Jr., J. M. Rodríguez, A. Bocos, M. Manzano. *Reduction of lateral and longitudinal oscillations of vehicles' platooning by means of decentralized overlapping control*. 46th IEEE Conference on Decision and Control. IEEE Catalog Number: 07CH37933C, ISBN: 1-4244-1498-9, New Orleans, Louisiana. USA. December 12-14, 2007.

- [Espinosa *et al.*, 2008a] F. Espinosa, M. Salazar, F. Valdés, A. Bocos. *Communication architecture based on Player/Stage and sockets for cooperative guidance of robotic units*. 16th Mediterranean Conference on Control and Automation, MED '08. Córcega, France, 2008. Proceedings. IEEE Catalog Number: CFP08MED-CDR, ISBN: 978-1-4244-2505-1, pp: 1423-142. 2008.
- [Espinosa *et al.*, 2008b] F. Espinosa, M. Salazar, A. Bocos, F. Valdés, R. Iglesias. *Design and implementation of a communication architecture based on Player/Stage for telerobotics operation of P3-DX units*. Int. Conference on Robotics and Automation, ICRA 2008. Pasadena, California, USA, 2008. IEEE Catalog Number: CFP08RAA-CDR, ISBN: 978-1-4244-1647-9, ISSN: 1050-4729. pp: 65-70. 2008.
- [Espinosa *et al.*, 2010] F. Espinosa, M. Salazar, D. Pizarro, F. Valdés. *Remote and Telerobotics*. Chapter title: *Electronics proposal for telerobotics operation of P3-DX units*. ISBN: 978-953-307-081-0. pp: 1 – 16. Marzo 2010. Editor: Nicolas Mollet Published by IN-TECH. Book Open Access: <http://www.sciyo.com/books/show/title/remote-and-telerobotics>. Chapter Open Acces: <http://www.sciyo.com/articles/show/title/electronics-proposal-for-telerobotics-operation-of-p3-dx-units>.
- [Espinosa *et al.*, 2011] F. Espinosa, C. Santos, M. Marrón-Romera, D. Pizarro, F. Valdés, J. Dongil. *Odometry and Laser Scanner Fusion Based on a Discrete Extended Kalman Filter for Robotic Platooning Guidance*. Journal Sensors-Basel 2011, 11, 8339-8357. Doi: 10.3390/s110908339. Área: Instruments & Instrumentation. Índice de impacto 1.821. Posición de la revista en el área 11/58.
- [EIS, 2012] Europe`s Information Society.  
[http://ec.europa.eu/information\\_society/newsroom/cf/menu.cfm](http://ec.europa.eu/information_society/newsroom/cf/menu.cfm)
- [Fan *et al.*, 2010] D. Fan. P. Shi. *Improvement of Dijkstra's algorithm and its application in route planning*. Seventh Int'l Conf. on Fuzzy Systems and Knowledge Discovery (FSKD), 2010 vol. 4. pp: 1901-1904, 2010. doi: 10.1109/FSKD.2010.5569452.
- [Ford, 1956] L. R. Ford, Jr. *Network flow theory*. Technical Report P-923, RAND, Santa Monica, CA, August, 14. 1956.
- [Furcy *et al.*, 2000] D. Furcy, S. Koenig. *Speeding up the Convergence of Real-Time Search*. Proceedings of the National Conference on Artificial Intelligence. 2000. Austin, TX. Pp: 891–897.
- [Gao, 2010] Y. Gao. *An Improved Shortest Route Algorithm in Vehicle Navigation System*. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). pp: V2-363 - V2-366. doi: 10.1109/ICACTE.2010.5579284.

- [Geintra, 2012] Grupo de Ingeniería Electrónica Aplicada a Espacios Inteligentes y Transporte. <http://www.geintra-uah.org/idi/demostraciones/demostraciones#robot-routing>
- [Ghoniem *et al.*, 2005] M. Ghoniem, J. D. Fekete, P. Castagliola. *On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis*. Information Visualization. 4(2). pp: 114-135. 2005.
- [GMV, 2010] GMV Informe Anual 2010: [www.gmv.com/es/Empresa/Publicaciones/InformeAnual/2010/index.html](http://www.gmv.com/es/Empresa/Publicaciones/InformeAnual/2010/index.html)
- [Goldberg *et al.*, 2004] A. V. Goldberg, Ch. Harrelson. *Computing the Shortest Path: A\* Search Meets Graph Theory*. Microsoft Research. Technical Report. MSR-TR-2004-24. 2004.
- [Golden, 1976] B. Golden. *Shortest-Path Algorithms: A Comparison*. Operations Research, Vol.44, pp: 1164-1168.
- [Golden *et al.*, 2001] B. L. Golden, A. A. Assad, E. A. Wasil. *Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries*. In: The Vehicle Routing Problem, Society for Industrial and Applied Mathematics. SIAM Monographs on Discrete Mathematics and Applications, pp: 245-286. SIAM. 2001.
- [Guggenheim, 1955] E. A. Guggenheim. *Boltzmann's Distribution Law*. North-Holland Publishing Comp. Amsterdam, 1955.
- [Halaoui, 2009] H. F. Halaoui. *Smart Traffic Systems: Dynamic A\* Traffic in GIS Driving Paths Applications*. World Congress on Computer Science and Information Engineering. pp: 626 - 630, 2009.
- [Hart *et al.*, 1968] P. E. Hart, N. J. Nilsson, B. Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics. Issue Date: July 1968. Vol. 4 Issue: 2. pp: 100 - 107. ISSN: 0536-1567.
- [Hashimoto *et al.*, 2008] H. Hashimoto, M. Yagiura, T. Ibaraki. *An iterated local search algorithm for the time-dependent vehicle routing problem with time windows*. Discrete Optimization, Vol. 5. pp: 434-456. 2008.
- [Hernández *et al.*, 2005] C. Hernandez, P. Meseguer. *Propagating updates in real-time search: FALCONS(k)*. 25th International Conference of the Chilean Computer Science Society, 2005. SCCC 2005. ISSN : 1522-4902. Print ISBN: 0-7695-2491-5.
- [Hoitner *et al.*, 2012] A. Hoitner, R. Herring, A. Bayen. *Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach*. 91st Transportation Research Board Annual Meeting, January 2012.



- [Hokuyo, 2012] Range-finder laser scanner URG-04LX.  
<http://www.acroname.com/robotics/parts/R283-HOKUYO-LASER1.html>
- [Horowitz *et al.*, 1978] E. Horowitz, S. Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press Inc., 1978.
- [Houghton *et al.*, 2009] J. Houghton, J. Reiners, C. Lim. *Transporte inteligente. Cómo mejorar la movilidad en las ciudades*. IBM Institute for Business Value. IBM Global Business. 2009.
- [Hu *et al.*, 1997] H. Hu, M. Brady. *Dynamic Global Path Planning with Uncertainty for Mobile Robots in Manufacturing*. IEEE Transactions on Robotics and Automation. Vol. 13 Issue: 5. pp: 760 - 767. ISSN: 1042-296X. 1997.
- [Hua *et al.*, 2010] M. Hua, J. Pei. *Probabilistic path queries in road networks: traffic uncertainty aware path selection*. Proceedings of the 13th International Conference on Extending Database Technology, March 22-26, 2010, Lausanne, Switzerland.
- [Huang *et al.*, 2007] B. Huang, Q. Wu, F. B. Zhan. *A shortest path algorithm with novel heuristics for dynamic transportation networks*. International Journal of Geographical Information Science, 21(6): 625-644, 2007.
- [Ishida *et al.*, 1996] T. Ishida, M. Shimbo. *Improving the learning efficiencies of real-time search*. In: Proc. AAAI-96, Vol. 1, Portland, OR, 1996, pp: 305-310.
- [Jefferis *et al.*, 1965] R. P. Jefferis, K. A. Fegley. *Application of Dynamic Programming to Routing Problems*. IEEE Transactions on Systems Science and Cybernetics. Vol. SSC-1 No. 1. 1965.
- [Jeong *et al.*, 2011] J. Jeong, T. He, D. Hu. *TMA: Trajectory-based Multi-Anycast for Multicast Data Delivery in Vehicular Networks*. Technical Report. Department of Computer Science and Engineering. University of Minnesota. 2011.
- [Jing *et al.*, 2009] M. Jing, Ch. Yu, H. Lee, L. Chen. *Solving Japanese Puzzles with Logical Rules and Depth First Search Algorithm*. Proceedings of the Eighth International Conference on Machine Learning and Cybernetics. Baoding, July 2009.
- [Jocksch, 1966] H. C. Jocksch. *The shortest route problem with constraints*. Journal of Math.Anal. Applicat., Vol. 14, pp: 191-197, 1966.
- [Joshi *et al.*, 2006] H. M. Joshi, J. A. Mcadams. *Search Algorithms in Intelligent Agents*. Scientific Paper on various search algorithms. Last referred on March 2006.
- [Jula *et al.*, 2006] H. Jula, M. Dessouky, P. A. Ioannou. *Truck route planning in nonstationary stochastic networks with time windows at customer locations*. IEEE Transactions on Intelligent Transportation Systems, Vol. 7 Issue:1. pp: 51 - 62. ISSN: 1524-9050.
- [Jula *et al.*, 2008] H. Jula, M. Dessouky, P.A. Ioannou. *Real-Time Estimation of Travel Times Along the Arcs and Arrival Times at the Nodes of Dynamic Stochastic*

- Networks*. IEEE Transactions on Intelligent Transportation Systems. Vol. 9. Issue: 1. pp: 97 - 110. 2008.
- [Kang *et al.*, 2007] J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J. Park, S. Won Bang. *Path Planning for Complete and Efficient Coverage Operation of Mobile Robots*. International Conference on Mechatronics and Automation, ICMA 2007. Aug. 2007., pp: 2126 - 2131. Harbin. Print ISBN: 978-1-4244-0828-3.
- [Kang *et al.*, 2008] H. Kang, B. Lee, K. Kim. *Path Planning Algorithm Using the Particle Swarm Optimization and the Improved Dijkstra Algorithm*. Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. PACIIA '08. Vol. 2. pp: 1002 - 1004. Wuhan. Print ISBN: 978-0-7695-3490-9.
- [Kaparias *et al.*, 2008] I. Kaparias, M. Bell, H. Belzner. *A new measure of travel time reliability for in-vehicle navigation systems*. Journal on Intelligent Transportation Systems. 12(4). pp: 202-211. Doi: 10.1080/15472450802448237. 2008.
- [Karimadini *et al.*, 2011] M. Karimadini, H. Lin. *Guaranteed global performance through local coordinations*. *Automatica*, Vol. In Press, Corrected Proof, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V21-529M7SH-5/2/19d8df66047d3ddc475af4b1b79ef8d2>.
- [Kharoufeh *et al.*, 2004] J. Kharoufeh, N. Gautam. *Deriving Link Travel-Time Distributions via Stochastic Speed Processes*. Transportation Science. Vol. 38, No. 1, February 2004, pp: 97-106. ISSN 0041-1655.
- [Khosroshahi *et al.*, 2011] A. H. Khosroshahi, P. Keshavarzi, Z. D. KoozehKanani, J. Sobhi. *Acquiring real time traffic information using VANET and dynamic route guidance*. IEEE 2nd International Conference on Computing, Control and Industrial Engineering (CCIE). Doi: 10.1109/CCIENG.2011.6007944. Vol 1. pp: 9-13. 2011.
- [Koenig *et al.*, 2002a] S. Koenig, M. Likhachev. *Incremental A\**. In Advances in Neural Information Processing Systems. MIT Press, 2002. pp: 1539-1546.
- [Koenig *et al.*, 2002b] S. Koenig, M. Likhachev. *D\* Lite*. Proceedings of the AAAI Conference on Artificial Intelligence. pp: 476-483. 2002.
- [Koenig *et al.*, 2002c] S. Koenig, M. Likhachev. *Improved Fast Replanning for Robot Navigation in Unknown Terrain*. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). 2002.
- [Kong *et al.*, 2011] D. Kong, Y. Liang, X. Ma, L. Zhang. *Improvement and Realization of Dijkstra Algorithm in GIS of Depot*. International Conference on Control, Automation and Systems Engineering (CASE), 2011.
- [Korf, 1985] R. E. Korf, *Depth-first iterative-deepening: An optimal admissible tree search*. Artificial Intelligence, 27(1), 97-109.
- [Korf, 1987] R. E. Korf. *Real-Time heuristic Search: First Results*. In Proceedings of the AAAI, pp: 133-138, 1987.

- [Korf, 1990] R. E. Korf. *Real-time heuristic search*. Artificial Intelligence, 42(2-3), pp: 189–211. 1990.
- [Kundu et al., 2005] D. Kundu, A. Manglick. *Discriminating between the log-normal and gamma distributions*. Journal of the Applied Statistical Sciences. Vol. 14, pp: 175–187. 2005.
- [Kwon et al., 2000] J. Kwon, B. Coifman, P. Bickel. *Day-to-Day travel time trends and travel-time prediction from loop-detector data*. Transportation Research Record (1717), 120-129. 2000.
- [Laporte, 2009] G. Laporte. *Fifty years of vehicle routing*. Transportation Science 43(4), 408–416. 2009.
- [Lee, 1961] C. Y. Lee. *An algorithm for path connections and its applications*. IRE Trans. on Electronic Computers, Vol. EC-10, pp: 346-365. 1961.
- [Leu et al., 2004] F. Y. Leu, Y. Ch. Fan. *Compressing a directed massive graph using small world model*. In Proceedings of the Conference on Data Compression, USA, 2004. IEEE Computer Society.
- [Levy et al., 2010] J. I. Levy, J. J. Buonocore, K. von Stackelberg. *Evaluation of the public health impacts of traffic congestion: a health risk assessment*. Environmental Health. 2010; 9: 65. Published online 2010 October 27. Doi: 10.1186/1476-069X-9-65. 2010.
- [Li, 2004] R. Li. *Examining Travel Time Variability using AVI Data*. Working Paper ITS-WP-04-23. Institute of Transport Studies. 2004. ISSN 1440 - 3501. [www.eng.monash.edu.au/civil/assets/document/research/centres/its/publications/its-wp-04-23.pdf](http://www.eng.monash.edu.au/civil/assets/document/research/centres/its/publications/its-wp-04-23.pdf)
- [Lin et al., 2008] Sh. Lin, W. Cai. *Logistics vehicle routing problem between two objects based on real-time traffic data*. IEEE International Conference on Service Operations and Logistics, and Informatics, IEEE/SOLI 2008. Issue Date: 12-15 Oct. 2008. pp: 2989 - 2994. Beijing. Print ISBN: 978-1-4244-2012-4.
- [Liu et al., 2011] X. Liu, D. Gong. *A comparative study of A-star algorithms for search and rescue in perfect maze*. International Conference on Electric Information and Control Engineering (ICEICE). pp: 24 - 27. Wuhan. Print ISBN: 978-1-4244-8036-4. April 2011.
- [Lu et al., 2011] Y. Lu, X. Huo, O. Arslan, P. Tsiotras. *Incremental Multi-Scale Search Algorithm for Dynamic Path Planning With Low Worst-Case Complexity*. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics. 2011.
- [Luger, 2005] G. F. Luger. *Artificial Intelligence. Structures and Strategies for Complex Problem Solving*. 6<sup>th</sup> Ed. ISBN-13: 978-0-321-54589-3.
- [Ma et al., 2009] W. Ma, Y. Xu, H. Xie. *The optimal path algorithm for emergency rescue for drilling accidents*. Proceedings of IC-NIDC. 2009.

- [Mahdi, 2011] A. Mahdi. *Estimation of Travel Time Reliability for Freeways Using Mean and Standard Deviation of Travel Time*. Journal of Transportation Systems Engineering and Information Technology. Vol. 11, Issue 6, December 2011.
- [Mainali et al., 2008] M.K. Mainali, K. Shimada, S. Mabu, K. Hirasawa. *Optimal route of road networks by dynamic programming*. IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). June 2008. pp: 3416 - 3420. Hong Kong. ISSN: 1098-7576. Print ISBN: 978-1-4244-1820-6. 2008.
- [MARTA, 2011] Movilidad y Automoción con Redes de Transporte Avanzadas. Ficosa Intl. [http://www.ficosa.com/wps/wcm/connect/ficosa/WebPublica\\_ES](http://www.ficosa.com/wps/wcm/connect/ficosa/WebPublica_ES).
- [Maue et al., 2006] J. Maue, P. Sanders, D. Matijevic. *Goal directed shortest path queries using precomputed cluster distances*. In 5th Workshop on Experimental Algorithms (WEA), Number 4007. IN LNCS, pp: 316–328. Springer, 2006.
- [Montgomery et al., 2003] D. C. Montgomery, G. C. Runger. *Applied Statistics and Probability for Engineers*. Third Edition. John Wiley & Sons. ISBN: 0471204544. 2003.
- [Mood et al., 1974] A. Mood, F. Graybill, D. Boes. *Introduction to the theory of statistics*. Third Edition. Mcgraw-Hill College. New York, ISBN: 0070428646. 1974.
- [Moore, 1959] E. F. Moore. *The shortest path through a maze*. Proceedings of International Symposium on the Theory of Switching, Part II, pp: 285-292. Harvard University Press, Cambridge, Massachusetts. 1959.
- [Nazari et al., 2008] S. Nazari, M. R. Meybodi, M. A. Salehigh, S. Taghipour. *An Advanced Algorithm for Finding Shortest Path in Car Navigation System*. First International Conference on Intelligent Networks and Intelligent Systems, 2008. ICINIS '08. Issue Date: 1-3 Nov.2008. pp: 671 - 674. Print ISBN: 978-0-7695-3391-9.
- [Nehmzov, 2006] U. Nehmzow. *Scientific methods in mobile robotics*, Quantitative analysis on agent behaviour, ISBN: 10-1-84628-019-2, Springer. 2006.
- [Nehmzov, 2008] U. Nehmzwo. *Robot Behaviour. Design, Description, Analysis and Modelling*. ISBN: 978-1-84800-396-5 e-ISBN: 978-1-84800-397-2. Doi 10.1007/978-1-84800-397-2. 2008.
- [Nie et al., 2012] Y. Nie, X. Wu, J. F. Dillenburg, P. C. Nelson. *Reliable route guidance: A case study from Chicago*. Transportation Research Part A 46 (2012) 403–419.
- [Nilsson, 1971] N. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill. 1971. ISBN: 0070465738.
- [Nilsson, 1998] N. Nilsson, *Artificial Inteligence: A New Synthesis*, Morgan Kaufmann Publishers, ISBN: 1-55860-535-5, 1998.

- [Noland *et al.*, 2002] R. B. Noland, J. W. Polak. *Travel Time Variability: A review of Theoretical and Empirical Issues*. Transport Reviews, Vol. 22 (1). pp: 39-54. ISSN: 0144-1647. 2002.
- [Noto *et al.*, 2000] M. Noto, H. Sato. *A Method for the Shortest Path Search by Extended Dijkstra Algorithm*. IEEE International Conference on Systems, Man, and Cybernetics, 2000. pp: 2316 - 2320 Print ISBN: 0-7803-6583-6.
- [Novoa *et al.*, 2009] C. Novoa, R. Storer. *An approximate dynamic programming approach for the vehicle routing problem with stochastic demands*. European Journal of Operational Research 196, 509–515. 2009.
- [Osvald *et al.*, 2008] A. Osvald, L. Zadnik. *A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food*. J. Food Eng. 85, 285–295 (2008).
- [Ou *et al.*, 2010] W. Ou, B. Sun. *A Dynamic Programming Algorithm for Vehicle Routing Problems*. International Conference on Computational and Information Sciences (ICCIS), 2010. doi: 10.1109/ICCIS.2010.182. pp: 733 - 736. 2010.
- [P3-DX, 2012] Pioneer P3-DX Datasheet, 2012. <http://www.mobilerobots.com/researchrobots/pioneerp3dx.aspx>
- [Pearl *et al.*, 1982] J. Pearl, J. H. Kim. *Studies in semi-admissible heuristics*. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-4(4):391–399. 1982.
- [Pearl, 1984] J. Pearl. *Heuristics*. Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley. 1984.
- [Player/Stage, 2012] The Player Project. <http://playerstage.sourceforge.net/>
- [Pohl, 1969] I. Pohl. *Bi-directional and heuristic search in path problems*. Tech. rep. 104, SLAC (Stanford Linear Accelerator Center, Stanford, California).
- [Pohl, 1970] I. Pohl. *Heuristic search viewed as path finding in a graph*. Artificial Intelligence, 1:193–204, 1970.
- [Pohl, 1973] I. Pohl. *The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computation issues in heuristic problem solving*. In Proceedings of IJCAI-73, 12–17. 1973.
- [Polus, 1979] A. Polus. *A study of travel time and reliability on arterial routes*. Transportation, Vol. 8, 1979, pp: 141-151.
- [Quintía *et al.*, 2010] P. Quintía, R. Iglesias, C. V. Regueiro, M. A. Rodríguez. *Simultaneous learning of perception and action in mobile robots*. Robotics and autonomous systems. Vol. 58. pp: 1306-1315. 2010.
- [Rakha *et al.*, 2006] H. Rakha, I. El-Shawarby, M. Arafah, F. Dion. *Estimating Path Travel-Time Reliability*. Proceedings of the IEEE ITSC 2006 Intelligent Transportation Systems Conference. Toronto, Canada, September 17-20, 2006.
- [Ramalingam *et al.*, 1996] G. Ramalingam, T.W. Reps. *An Incremental Algorithm for a Generalization of the Shortest-Path Problem*. J. Algorithms, Vol. 21, no. 2, pp: 267-305. 1996.

- [RITA, 2011] RITA. *U.S. Department of Transportation Research and Innovative Technology Administration*. [www.its.dot.gov](http://www.its.dot.gov)
- [Rodríguez *et al.*, 2006] J. M. Rodríguez, A. M.H. Awawdeh, F. Espinosa, J. Pastor, F. Valdés, M. A. Ruiz, A. Gil. *Electronic solution based on micro-controller AT91SAM7S256 for platooning multi-agent system implementation*. 2nd International Conference on Informatics in Control, Automation and Robotics - ICINCO 2006-ISBN: 978-972-8865-60-3. pp: 202-209. Setúbal. Portugal. 2006.
- [Rodríguez *et al.*, 2009] M. Rodríguez, R. Iglesias, F. Espinosa, P. Quintía, C. V. Regueiro, F. Valdés. *Learning proposal based on reinforcement for collaborative tasks: robot convoy formation*. Proceedings of the 4rd European Conference on Mobile Robotics (ECMR 09). ISBN: 978-953-6037-54-4. pp: 147-154. Dubrovnik, Croacia, 23-25 Septiembre 2009.
- [Russell *et al.*, 2003] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. Second Edition. Prentice Hall. 2003.
- [Ruzika *et al.*, 2011] S. Ruzika, M. Thiemann. *Reliable and Restricted Quickest Path Problems*. INOC 2011, 9961, p.1-8. 2011.
- [SAFESPOT Project, 2011] *Co-operative Systems for Road Safety "Smart Vehicles on Smart Roads"*. 2011. <http://www.safespot-eu.org/>
- [Saizheng *et al.*, 2009] W. Saizheng, L. Kejun. *Vehicle Routing Algorithm Based on Real-Information*. Second International Conference on Intelligent Computation Technology and Automation, 2009. ICICTA '09. Vol. 3. Doi: 10.1109/ICICTA.2009.678.
- [Salazar, 2011] M. Salazar. *Integración de Matlab/Simulink/RTW y Linux/RTAI para aplicaciones de control y comunicaciones en robótica*. Trabajo Fin de Carrera. Ingeniería en Electrónica. Universidad de Alcalá. Marzo 2011.
- [Sanders *et al.*, 2007] P. Sanders, D. Schultes. *Engineering fast route planning algorithms*. WEA 2007. Vol. 4525, pp: 23–36. Springer, Heidelberg. 2007.
- [Santos *et al.*, 2010] C. Santos, F. Espinosa, D. Pizarro, F. Valdés, E. Santiso, I. Díaz. *Fuzzy decentralized control for guidance of a convoy of robots in non-linear trajectories*. 15th IEEE Int. Conf. on Emerging Technologies and Factory Automation -ETFA'10-.Bilbao, Spain. ISBN: 978-1-4244-6849-2. 2010.
- [Santos, 2010] C. Santos. *Propuesta de control descentralizado con solapamiento para guiado en convoy de unidades P3-DX*. Trabajo Fin de Carrera. 2010.
- [Sedgewick *et al.*, 1986] R. Sedgewick, J. Vitter. *Shortest Paths in Euclidean Graphs*. Algorithmica. Vol. 1 (1-4). pp: 31-48. Doi: 10.1007/BF01840435. 1987.
- [Shimbo *et al.*, 2003] M. Shimbo, T. Ishida. *Controlling the learning process of real-time heuristic search*. Artificial Intelligence, 146(1), 1–41. Doi:10.1016/S0004-3702(03)00012-2. 2003.

- [Song *et al.*, 2010] Q. Song, X. Wang. *Efficient Routing on Large Road Networks Using Hierarchical Communities*. IEEE Transactions on Intelligent Transportation Systems, Vol. 12 Issue:1. pp: 132 - 140. ISSN: 1524-9050. 2010.
- [Stentz, 1994] A. Stentz. *Optimal and Efficient Path Planning for Partially-Known Environments*. Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Vol. 4, pp: 3310 -3317. 1994.
- [Stentz, 1995] A. Stentz. *The Focussed D\* Algorithm for Real-Time Replanning*. Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Quebec. 1995.
- [Stentz, 1996] A. Stentz. *Map-Based Strategies for Robot Navigation in Unknown Environments*. Proc. AAAI Spring Symp. Planning With Incomplete Information for Robot Problems, pp: 110 -116. 1996.
- [Sun *et al.*, 2003] C. Sun, G. Arr, R. P. Ramachandran. *Vehicle reidentification as method for deriving travel time and travel time distribution*. Transportation Research Record (1826), 25-31. 2003.
- [Sung *et al.*, 2000] K. Sung, M. G. Bell, M. Seong, S. Park. *Shortest paths in a network with time-dependent flow speeds*. European Journal of Operational Research, 2000, 121(12), pp: 32-39.
- [Sutton *et al.*, 1998] R. Sutton, A. G. Barto. *Reinforcement Learning, An introduction*. A Bradford Book, The MIT Press, 1998.
- [Talley *et al.*, 1987] W. K. Talley, A. J. Becker. *On-time performance and the exponential probability distribution*. Transportation Research Record. Issue 1108. pp: 22-26. ISSN: 0361-1981. ISBN: 0-309-04456-1. 1987.
- [Thayer *et al.*, 2008] J. Thayer, W. Ruml. *Faster Than Weighted A\*: An Optimistic Approach to Bounded Suboptimal Search*. In Proc. ICAPS, 2008, pp: 355-362.
- [Thayer *et al.*, 2010] J. Thayer, W. Ruml. *Finding Acceptable Solutions Faster Using Inadmissible Information*. Computer Science Technical Report 10-01.
- [SARTRE, 2011] The SARTRE Project. *Safe Road Trains for the Environment*. <http://www.sartre-project.eu/en/Sidor/default.aspx>
- [Toor *et al.*, 2008] Y. Toor, P. Muhlethaler, A. Laouiti, A. De La Fortelle. *Vehicle ad hoc networks: Applications and related technical issues*. IEEE Communications Surveys & Tutorials, 10(3). pp: 74–88, 2008.
- [Trinh *et al.*, 2011] K. Trinh, N. Dang, T. Dinh. *An Approximation Approach for a Real-World Variant of Vehicle Routing Problem*. 3rd Asian Conference on Intelligent Information and Database Systems (ACIIDS2011), 2011.
- [Valdés *et al.*, 2006] F. Valdés, F. Espinosa, A. M. H. Awawdeh, J. Pastor. *Routes for splitting and merging maneuvers of platoon followers in urban environments*. 11th IEEE International Conference on Emerging Technologies and Factory

- Automation, Proceedings ETFA 2006. República Checa, 2006. ISBN: 1-4244-0681-1, pp: 1153-1156.
- [Valdés *et al.*, 2008] F. Valdés, F. Espinosa, M. Salazar, J. Calderón. *Fuzzy decision system implementation for leader trajectory discretizer of robotic units in convoy*. International Conference FIE 2008. Conference Proceedings. Santiago de Cuba, Cuba. 2008.
- [Valdés *et al.*, 2009] F. Valdés, R. Iglesias, F. Espinosa, M. A. Rodríguez. *Propuesta de estrategia para el enrutamiento adaptable aplicado a sistemas de navegación en entornos urbanos*. IX Congreso Español de Sistemas Inteligentes de Transporte. Andorra La Vella. Andorra, 2009.
- [Valdés *et al.*, 2011a] F. Valdés, R. Iglesias, F. Espinosa, M. A. Rodríguez, P. Quintia, C. Santos. *Robot Routing Approaches for Convoy Merging Manoeuvres*. 12th Conference Towards Autonomous Robotic Systems. TAROS 2011, August 31 - September 2, 2011. Sheffield, UK.
- [Valdés *et al.*, 2011b] F. Valdés, R. Iglesias, F. Espinosa, M. A. Rodríguez. *An Efficient Algorithm for Optimal Routing Applied to Convoy Merging Maneuvers in Urban Environments*. Journal on Applied Intelligence. Springer. November 2011. Springer. DOI 10.1007/s10489-011-0326-8.
- [Valdés *et al.*, 2011c] F. Valdés, R. Iglesias, F. Espinosa, M. A. Rodríguez, P. Quintía and C. Santos. Lecture Notes in Artificial Intelligence. *Towards Autonomous Robotic Systems*. Chapter Title: *Robot routing approaches for convoy merging maneuvers*. ISSN: 0302-9743. ISBN 978-3-642-23231-2. Clave: CL, Vol. 6856, pp: 241- 252. Sept. 2011. Springer-Verlag. Doi 10.1007/978-3-642-23232-9.
- [Valdés *et al.*, 2012a] F. Valdés, R. Iglesias, F. Espinosa, M.A. Rodríguez, P. Quintía, C. Santos. *Implementation of robot routing approaches for convoy merging manoeuvres*. Robotics and Autonomous Systems (2012), doi:10.1016/j.robot.2012.05.003.
- [Valdés *et al.*, 2012b] F. Valdés, F. Espinosa, R. Iglesias, Miguel A. Rodriguez. *Effect of a risk factor in convoy merging manoeuvres considering uncertainty in travelling times*. Applied Soft Computing, Elsevier, Segunda revisión enviada en abril de 2012.
- [VIA-T, 2011] *Sistema de Tele-peaje*. <http://www.viat.es/>
- [VISNU, 2011] VISNU Project. Identification and Interaction of Multiple Agents in Intelligent Spaces Using a Camera Array. Departamento de Electrónica. Universidad de Alcalá. 2012. <http://www.geintra-uah.org/projects/visnu>
- [Wagner *et al.*, 2005] D. Wagner, T. Willhalm, C. Zaroliagis. *Geometric containers for efficient shortest-path computation*. ACM Journal of Experimental Algorithmics 10, 1–30 (2005).



- [Wagner *et al.*, 2007] D. Wagner and T. Willhalm. *Speed-up techniques for shortest-path computations*. In W. Thomas and P. Weil, editors, 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2007), Vol. 4393 of LNCS. pp: 23–36, New York, 2007. Springer.
- [Wang *et al.*, 2011] H. Wang, Q. Yang. *Application of Dijkstra algorithm in robot path-planning*. Second International Conference on Mechanic Automation and Control Engineering (MACE), 2011. Issue Date: 15-17 July 2011. pp: 1067 - 1069. Print ISBN: 978-1-4244-9436-1.
- [Watkins, 1989] C. Watkins. *Learning from delayed rewards*. Ph.D. Thesis, University of Cambridge, England, 1989.
- [Willms *et al.*, 2006] A. R. Willms, S. X. Yang. *An Efficient Dynamic System for Real-Time Robot-Path Planning*. IEEE Transactions on Systems, Man, and Cybernetics. Part B. Vol. 36, No. 4, August 2006.
- [Xi *et al.*, 2006] Ch. Xi, F. Qi, L. Wei. *A New Shortest Path Algorithm based on Heuristic Strategy*. The Sixth World Congress on Intelligent Control and Automation, 2006. WCICA 2006. pp: 2531 - 2536. Print ISBN: 1-4244-0332-4.
- [Xu *et al.*, 2008] L. Xu, Z. Gao. *Bi-objective urban road transportation discrete network design problem under demand and supply uncertainty*. IEEE International Conference on Automation and Logistics, 2008. ICAL 2008.
- [Yang *et al.*, 2010] Y. Yang, Z. Mi, J. Yifei, G. Liu. *A Model Based Connectivity Improvement Strategy for Vehicular Ad hoc Networks*. IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), 2010.
- [Yu *et al.*, 2009] S. Yu, S. Mabu, F. Ye, H. Wang, K. Shimada, K. Hirasawa. *Q Value-Based Dynamic Programming with Boltzmann Distribution for Global Optimal Traffic Routing Strategy*. Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 13, No. 5, pp: 581-591, Sep. 2009.
- [Zenital, 2011] GPS and Mobility Trades S. L. <http://www.zenital.es/planificacion.html>
- [Zhan *et al.*, 2000] F. B. Zhan, C. E. Noon. *A Comparison Between Label-Setting and Label-Correcting Algorithms for Computing One-to-One Shortest Paths*. Journal of Geographic Information and Decision Analysis, 4, 2000.
- [Zhou *et al.*, 2006] R. Zhou, E. Hansen. *Breadth-first heuristic search*. Artificial Intelligence 170 (4-5). pp: 385 – 408. 2006.
- [Zhu *et al.*, 2010] H. Zhu, G. Singh. *A Communication Protocol for a Vehicle Collision Warning System*. Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, ISBN: 978-0-7695-4331-4, pp: 636-644, 2010.

