



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

## *Optimización de recursos intralogísticos en entornos industriales para su uso en vehículos autónomos*

**Elliot Vargas Martín**

**ADVERTIMENT** La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (<http://upcommons.upc.edu/tesis>) i el repositori cooperatiu TDX (<http://www.tdx.cat/>) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA** La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (<http://upcommons.upc.edu/tesis>) y el repositorio cooperativo TDR (<http://www.tdx.cat/?locale-attribute=es>) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing*). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING** On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (<http://upcommons.upc.edu/tesis>) and the cooperative repository TDX (<http://www.tdx.cat/?locale-attribute=en>) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (*framing*). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de doctorado:

AUTOMÀTICA, ROBÒTICA Y VISIÒN

TESIS DOCTORAL

**OPTIMIZACIÒN DE RECURSOS  
INTRALOGÍSTICOS EN ENTORNOS  
INDUSTRIALES PARA SU USO EN VEHÍCULOS  
AUTÓNOMOS**

Elliot Vargas Martín

Director: Manel Velasco García

Codirector: Pau Martí Colom

Junio de 2023



## Resumen

La introducción de vehículos autónomos en la intralogística ha aumentado significativamente en los últimos tiempos. Sin embargo, en la industria, a menudo no se utiliza un solo vehículo autónomo, sino que se necesitan varios de ellos para llevar a cabo los procesos logísticos. Esto ha llevado a la necesidad de implementar sistemas de Gestión de flotas para coordinar los vehículos de manera efectiva, mejorando así el rendimiento y la seguridad. La Gestión de flotas es la administración de un grupo de vehículos de transporte reunidos bajo un mismo propósito por una organización, y su objetivo principal es aumentar el rendimiento de los vehículos que controlan según las necesidades de cada empresa.

**El objetivo principal de esta tesis doctoral es reducir costes en el entorno intralogístico mediante el desarrollo de un algoritmo de Gestión de flotas de vehículos autónomos.** Para alcanzar este objetivo, primero se lleva a cabo una evaluación de diferentes algoritmos de Gestión de flotas a través del diseño de un simulador capaz de replicar el movimiento de los vehículos autónomos, con el fin de determinar los aspectos que pueden reducir los costes de una flota de vehículos autónomos.

El resultado de la evaluación de los algoritmos de Gestión de flotas determina que el rendimiento de un algoritmo se ve alterado por el lugar de actuación. Para abordar este problema y poder reducir los costes en la Gestión de flotas, se plantea **la hipótesis que algoritmos adaptativos de Gestión de flotas que respondan a las características específicas de cada entorno, permitirían una mejora significativa en el rendimiento global de la flota de vehículos gestionada por el sistema de Gestión de flotas, al adaptarse de manera eficiente a las condiciones específicas del entorno en cuestión.**

En consecuencia, se propone un algoritmo para generar otros algoritmos de Gestión de flotas específicamente adaptados al entorno, centrándose en la gestión del tráfico en las intersecciones y el cálculo de rutas. Siendo el objetivo de esta adaptación maximizar el rendimiento de la Gestión de flotas según los criterios de evaluación de cada entorno.

El algoritmo utiliza un enfoque de aprendizaje genético para adaptar el algoritmo de Gestión de flotas a la instalación, considerando los vehículos, el mapa y las tareas. Para realizar la evaluación de la adaptación, se desarrolla un algoritmo de emulación de flotas de vehículos, considerando varios aspectos como su dinámica, los sensores perimetrales, las decisiones del gestor de intersecciones y la generación de rutas. Este algoritmo de emulación de flotas de vehículos se utilizará adicionalmente para predecir la posición futura de los vehículos, lo que

permite tener en cuenta el tráfico en el cálculo de rutas y la gestión del tráfico.

Una vez generado el algoritmo de Gestión de flotas adaptado al entorno, este cuenta con un gestor de tráfico en las intersecciones optimizado para maximizar el rendimiento de los vehículos según los criterios de evaluación específicos de cada entorno. Además, el generador de rutas propuesto tiene en cuenta el tráfico generado y las decisiones del gestor de tráfico en las intersecciones. Esto se realiza mediante el uso de ventanas de tiempo, donde se añade al cálculo de una ruta, el retraso generado por el tráfico en cada localización y tiempo específico, utilizando la predicción de la posición generada previamente.

Finalmente, para validar que se cumple con el objetivo de la tesis, se compara el rendimiento de los algoritmos de Gestión de flotas generados en diferentes casos de uso, incluyendo un caso real, con el de otros algoritmos que no se adaptan al entorno. La validación se realiza mediante una prueba de hipótesis, donde se contrasta la diferencia de rendimiento entre los algoritmos adaptados y los no adaptados.

La conclusión de la tesis sostiene que **el empleo de un algoritmo generador de Gestores de flotas, capaz de crear algoritmos específicamente adaptados a un entorno dado, logra maximizar el rendimiento de los vehículos autónomos y, consecuentemente, reduce los costes asociados a la Gestión de flotas.**

## Abstract

The introduction of autonomous vehicles in intralogistics has significantly increased in recent times. However, in the industry, a single autonomous vehicle is often not used, but several of them are needed to carry out logistical processes. This has led to the need for implementing Fleet management systems to effectively coordinate the vehicles, thereby improving performance and safety. Fleet management involves overseeing group of transport vehicles gathered under the same purpose by an organization, and its main goal is to increase the performance of the controlled vehicles according to the needs of each company.

**The main objective of this doctoral thesis is to reduce costs in the intralogistics environment through the development of an algorithm for Fleet management of autonomous vehicles.** To achieve this objective, an evaluation of different Fleet management algorithms is first carried out through the design of a simulator capable of replicating the movement of autonomous vehicles, in order to determine the aspects that can reduce the costs of an autonomous vehicle fleet.

The result of the evaluation of the Fleet management algorithms determines that the performance of an algorithm is affected by the place of operation. To address this problem and be able to reduce costs in Fleet management, **the hypothesis is proposed that adaptive Fleet management algorithms that respond to the specific characteristics of each environment would allow a significant improvement in the overall performance of the fleet of vehicles managed by the Fleet management system, by efficiently adapting to the specific conditions of the environment in question.**

Consequently, an algorithm is proposed to generate other Fleet management algorithms specifically adapted to the environment, focusing on traffic management at intersections and route calculation. The objective of this adaptation is to maximize the performance of Fleet management according to the evaluation criteria of each environment.

The algorithm uses a genetic learning approach to adapt the Fleet management algorithm to the facility, considering the vehicles, the map, and the tasks. To perform the adaptation evaluation, a fleet emulation algorithm is developed, considering various aspects such as its dynamics, perimeter sensors, intersection manager decisions, and route generation. This fleet emulation algorithm will also be used to predict the future position of the vehicles, allowing traffic to be taken into account in route calculation and traffic management.

Once the Fleet management algorithm adapted to the environment is generated, it has a traffic manager at intersections optimized to maximize vehicle performance according to the specific evaluation criteria of each environment. In addition, the proposed route generator takes into account the generated traffic and the intersection traffic manager decisions. This is done by using time windows, where the delay generated by traffic at each specific location and time is added to the route calculation, using the previously generated position prediction.

Finally, to validate that the objective of the thesis is met, the performance of the Fleet management algorithms generated in different use cases, including a real case, is compared with that of other algorithms that do not adapt to the environment. The validation is done through a hypothesis test, where the performance difference between the adapted and non-adapted algorithms is contrasted.

The conclusion of the thesis maintains that **the use of a Fleet managers generation algorithm, capable of creating algorithms specifically adapted to a given environment, achieves maximum performance of autonomous vehicles and, consequently, reduces the costs associated with Fleet management.**

# Índice

<b>1</b>	<b>Motivación de la tesis, objetivos y estructura de la memoria</b>	<b>1</b>
1.1	Motivación . . . . .	2
1.2	Objetivos y contribuciones . . . . .	6
1.3	Estructura de la memoria . . . . .	8
<b>2</b>	<b>Sistemas de gestión de flotas en el entorno intralogístico</b>	<b>13</b>
2.1	Introducción . . . . .	14
2.2	Definición del entorno . . . . .	17
2.2.1	El espacio de las instalaciones . . . . .	18
2.2.2	Medios . . . . .	19
2.2.3	El trabajo a realizar . . . . .	21
2.3	Restricciones . . . . .	24
2.3.1	Restricciones de la industria . . . . .	24
2.3.2	Restricciones legislativas . . . . .	25
2.3.3	Restricciones del entorno . . . . .	26
2.3.4	Restricciones de los AGV . . . . .	26
2.4	Criterios de minimización de costes . . . . .	27
<b>3</b>	<b>Metodología de cálculo de rendimiento máximo de una flota de AGV</b>	<b>31</b>
3.1	Distribución de los AGV . . . . .	34
3.2	Cálculo de rendimientos sin restricción de tráfico . . . . .	35
3.2.1	Metodologías de cálculo analítico . . . . .	35
3.2.2	Metodologías de cálculo numérico mediante programas genéricos . . . . .	39
3.3	Algoritmo de cálculo dinámico del tiempo de trayecto . . . . .	43
3.3.1	Generación del grafo . . . . .	44
3.3.2	Resolución del grafo . . . . .	45
3.3.3	Extracción del tiempo de navegación . . . . .	47
3.3.4	Cálculo del tiempo de trayecto de un segmento que une dos nodos . . . . .	47
3.3.5	Ejemplo de extracción del tiempo de navegación . . . . .	50

3.4	Cálculos con restricción de tiempo de operación y carga de baterías	52
3.5	Extracción del rendimiento máximo mediante cálculo . . . . .	57
<b>4</b>	<b>Cálculo de rendimiento de los Gestores de flotas mediante simulación</b>	<b>59</b>
4.1	Características y restricciones del simulador . . . . .	61
4.2	Datos requeridos . . . . .	62
4.2.1	Los modelos de elementos industriales . . . . .	62
4.2.2	El espacio de las instalaciones . . . . .	64
4.2.3	El trabajo . . . . .	68
4.2.4	Definición de la flota . . . . .	71
4.2.5	Características de la simulación . . . . .	72
4.3	Datos de salida del simulador . . . . .	73
4.4	Metodología de cálculo de rendimiento máximo mediante simulador	75
<b>5</b>	<b>Estado del arte</b>	<b>77</b>
5.1	Introducción . . . . .	79
5.2	Algoritmo de límite de rendimiento . . . . .	81
5.3	Algoritmos convencionales . . . . .	81
5.4	Algoritmos no convencionales . . . . .	83
5.5	Estado del arte en metodologías de aprendizaje . . . . .	84
<b>6</b>	<b>Entornos a evaluar</b>	<b>87</b>
6.1	Introducción . . . . .	88
6.2	Entorno 1 . . . . .	89
6.2.1	AGV del entorno 1 . . . . .	90
6.2.2	Cuantificación del entorno 1 . . . . .	90
6.3	Entorno 2 . . . . .	91
6.3.1	AGV del entorno 2 . . . . .	92
6.3.2	Cuantificación del entorno 2 . . . . .	93
6.4	Entorno 3 . . . . .	94
6.4.1	AGV del entorno 3 . . . . .	96
6.4.2	Cuantificación del entorno 3 . . . . .	96
6.5	Entorno Real . . . . .	98
6.5.1	AGV del entorno real . . . . .	100
6.5.2	Cuantificación del entorno real . . . . .	101
<b>7</b>	<b>Análisis de rendimiento de los algoritmos de referencia</b>	<b>105</b>
7.1	Rendimiento de los algoritmos de Gestión de flotas en cada entorno	107
7.1.1	Entorno 1 . . . . .	107
7.1.2	Entorno 2 . . . . .	111
7.1.3	Entorno 3 . . . . .	114
7.2	Carga computacional de los algoritmos de Gestión de flotas en cada entorno . . . . .	117
7.3	Análisis de los algoritmos de Gestión de flotas . . . . .	118
7.3.1	<i>STPRAN</i> y <i>STPFIFO</i> . . . . .	119
7.3.2	<i>STPACC</i> . . . . .	119

7.3.3	<i>STPCT</i> . . . . .	119
7.3.4	<i>AStar</i> . . . . .	120
7.3.5	<i>GEOPAR</i> . . . . .	120
7.3.6	<i>TIMEWINDOW</i> . . . . .	121
7.4	Comparativa del rendimiento de los diferentes algoritmos de referencia . . . . .	122
7.5	La mejor combinación . . . . .	125
<b>8</b>	<b>Algoritmo generador de gestores de flotas: Fase I - Entorno simplificado</b>	<b>127</b>
8.1	Definición del entorno . . . . .	129
8.1.1	Definición de la intersección . . . . .	129
8.1.2	Definición del entorno de evaluación . . . . .	130
8.1.3	Cuantificación . . . . .	131
8.2	Algoritmos de referencia . . . . .	131
8.3	Algoritmo GA1 . . . . .	132
8.3.1	Aprendizaje . . . . .	134
8.3.2	Modelización del entorno . . . . .	141
8.3.3	Extracción de los datos . . . . .	143
8.3.4	Fase de selección . . . . .	145
8.3.5	Fase de comprobación . . . . .	146
8.3.6	Algoritmo completo . . . . .	149
8.4	Resultados . . . . .	151
8.4.1	Resultados del aprendizaje . . . . .	153
8.4.2	Análisis de los resultados . . . . .	153
<b>9</b>	<b>Algoritmo generador de gestores de flotas: Fase II - Entorno completo</b>	<b>155</b>
9.1	Definición del entorno . . . . .	157
9.2	Algoritmos de referencia . . . . .	157
9.3	Algoritmo GA2 . . . . .	157
9.3.1	Metodología de aprendizaje . . . . .	158
9.3.2	Extracción de los datos . . . . .	163
9.3.3	Algoritmo completo . . . . .	165
9.4	Resultados . . . . .	167
9.4.1	Comparativa con los algoritmos de referencia . . . . .	168
9.4.2	Carga computacional . . . . .	169
9.4.3	Aprendizaje . . . . .	170
9.4.4	Análisis de los resultados . . . . .	171
<b>10</b>	<b>Algoritmo generador de gestores de flotas: Fase III - Algoritmo completo</b>	<b>173</b>
10.1	Definiciones del entorno . . . . .	175
10.1.1	El Roadmap . . . . .	175
10.1.2	El AGV . . . . .	176
10.1.3	Rutas . . . . .	177
10.1.4	Intersecciones . . . . .	178
10.2	Cálculo de la posición futura de los AGV . . . . .	178
10.2.1	Datos de inicialización . . . . .	179

10.2.2	Método de cálculo de la posición futura de los AGV . . . .	180
10.2.3	Cálculo dinámico de la movilidad de los AGV . . . . .	183
10.2.4	Emulación de la gestión del tráfico en las intersecciones .	194
10.2.5	Detención causada por el tráfico . . . . .	194
10.3	Cálculo mejorado de rutas . . . . .	203
10.3.1	Generación del grafo . . . . .	205
10.3.2	Resolución del grafo . . . . .	206
10.3.3	Cálculo de un segmento . . . . .	210
10.3.4	Extracción del camino más corto . . . . .	214
10.3.5	Generación de la ruta . . . . .	217
10.4	Prioridad de los AGV . . . . .	218
10.5	Gestión de intersecciones . . . . .	219
10.5.1	Aprendizaje . . . . .	219
10.5.2	Extracción de los datos . . . . .	225
10.5.3	Fase de selección . . . . .	231
10.5.4	Fase de comprobación . . . . .	232
10.5.5	Algoritmo completo de gestión de intersecciones . . . . .	232
10.6	Algoritmo GA3 completo . . . . .	235
<b>11</b>	<b>Evaluación del algoritmo propuesto</b>	<b>239</b>
11.1	Resultados en los tres entornos simulados . . . . .	241
11.1.1	Entorno y cuantificación . . . . .	241
11.1.2	Algoritmos de referencia . . . . .	241
11.1.3	Comparativa entre algoritmos en el entorno 1 . . . . .	242
11.1.4	Comparativa entre algoritmos en el entorno 2 . . . . .	244
11.1.5	Comparativa entre algoritmos en el entorno 3 . . . . .	245
11.1.6	Carga computacional . . . . .	247
11.2	Resultados en entorno de laboratorio real . . . . .	250
11.2.1	Tarea T3_1-3 y T3_3-1 . . . . .	254
11.2.2	Tareas T4_3-2 y T4_1-3 . . . . .	256
11.2.3	Tareas T9_3-2 y T9_2-1 . . . . .	257
11.3	Análisis de los resultados . . . . .	258
11.4	Prueba de hipótesis . . . . .	261
11.4.1	Metodología de cálculo la prueba de hipótesis . . . . .	261
11.4.2	Resultados de prueba de hipótesis . . . . .	263
<b>12</b>	<b>Conclusiones</b>	<b>265</b>
	<b>Glossary</b>	<b>269</b>
	<b>Bibliography</b>	<b>289</b>

## Datos Personales

**Autor:** Elliot Vargas Martín

**Universidad:** Universitat Politècnica de Catalunya

**Programa de doctorado:** ARV (Doctorado en Automática, Robótica y Visión)

**Director:** Dr. Manel Velasco García

**E-mail** manel.velasco@upc.edu

**Universidad:** Universitat Politècnica de Catalunya

**Departamento:** Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial

**Codirector:** Dr. Pau Martí Colom

**E-mail** pau.marti@upc.edu

**Universidad:** Universitat Politècnica de Catalunya

**Departamento:** Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial



# Lista de figuras

1.1.1	Estructura típica de control de un Gestor de flotas . . . . .	4
1.1.2	Estructura propuesta de generación y control de un Gestor de flotas específica para un entorno . . . . .	5
1.3.1	Estructura de la memoria en relación con los subobjetivos y objetivo de esta tesis . . . . .	9
2.2.1	Ejemplo de un AGV . . . . .	19
2.2.2	Ejemplo de entornos industriales . . . . .	21
2.2.3	Ejemplo de ejecución de una Tarea y sus Misiones . . . . .	23
3.2.1	Esquema básico de la realización de una tarea . . . . .	35
3.2.2	Esquema básico de ruta cíclica . . . . .	36
3.2.3	Esquema de ruta de un AGV en configuración con punto de salida único . . . . .	37
3.2.4	Esquema de ruta un AGV en configuración con punto de salida múltiple . . . . .	37
3.2.5	Esquema de ruta un AGV en configuración con distribución libre . . . . .	38
3.2.6	Elementos que determinan el comportamiento del <i>software</i> de simulación de eventos discretos en el caso de una tarea . . . . .	39
3.2.7	Bloques del <i>software</i> de simulación de eventos discretos para simular el caso cíclico sin restricciones . . . . .	40
3.2.8	Bloques del <i>software</i> de simulación de eventos discretos para simular el caso de punto de salida único sin restricciones . . . . .	41
3.2.9	Comparativa del resultado del número de AGV necesarios entre las <i>metodologías de cálculo analítico</i> y las <i>metodologías de cálculo numérico</i> (Valor más bajo mejor) . . . . .	42
3.3.1	Diagrama ejemplo de un <i>GrafoMapaResol</i> del entorno . . . . .	44
3.3.2	Diagrama del algoritmo de resolución del <i>GrafoMapaResol</i> sin restricción del tráfico . . . . .	45
3.3.3	Diagrama ejemplo de la resolución del grafo de la ruta . . . . .	50

4.2.1	Ejemplo de definición de un cargador de baterías de un AGV .	64
4.2.2	Ejemplo de definición de un elemento a transportar . . . . .	64
4.2.3	Ejemplo de definición de un <i>Roadmap</i> . . . . .	65
4.2.4	Detalle de ejemplo de definición de un <i>Roadmap</i> . . . . .	66
4.2.5	Detalle del un ejemplo de definición de puntos de entrega multiple	67
4.2.6	Información requerida para definir los puntos de carga de batería que cuenta una instalación . . . . .	68
4.2.7	Ejemplo de definición de unas tareas dentro de un entorno . .	69
4.2.8	Ejemplo de definición de una tarea . . . . .	69
4.2.9	Ejemplo de definición de un destino de una tarea . . . . .	71
4.2.10	Ejemplo de definición de una flota de AGV . . . . .	71
4.2.11	Ejemplo de configuración de una simulación . . . . .	72
4.2.12	Ejemplo de la asignación de posición inicial de una flota de AGV	73
4.3.1	Ejemplo de la visualización 3D resultante de una simulación .	73
4.3.2	Ejemplo de la visualización 3D resultante de una simulación, con la cámara cerca de los AGV . . . . .	75
6.2.1	Entorno industrial número 1 . . . . .	89
6.2.2	AGV modelo K10 de la empresa Kivnon, utilizado en el entor- no 1 . . . . .	90
6.3.1	Entorno industrial número 2 . . . . .	91
6.3.2	AGV modelo K05 de la empresa Kivnon, utilizado en el entor- no 2 . . . . .	92
6.4.1	Entorno industrial número 3 . . . . .	94
6.4.2	Visualización del entorno industrial con un motor gráfico en 3D	95
6.4.3	Definición de los puntos de entrega . . . . .	96
6.4.4	AGV modelo K11 de la empresa Kivnon, utilizado en el entor- no 3 . . . . .	96
6.5.1	Imagen tomada durante las pruebas en un entorno de labora- torio real . . . . .	98
6.5.2	Entorno de laboratorio real . . . . .	99
6.5.3	Visualización del entorno de laboratorio real con un motor gráfico en 3D . . . . .	100
6.5.4	AGV modelo K41 de la empresa Kivnon, utilizado en el entor- no real . . . . .	100
6.5.5	AGV modelo K55 de la empresa Kivnon, utilizado en el entor- no real . . . . .	101
6.5.6	Posibles tareas para el entorno real . . . . .	102
6.5.7	Tareas que forman parte del entorno real . . . . .	102
7.1.1	Resultado de los algoritmos de referencia del entorno 1 . . . .	107
7.1.2	Resultado de los algoritmos de referencia del entorno 2 . . . .	111
7.1.3	Resultado de los algoritmos de referencia del entorno 3 . . . .	114
7.2.1	Carga computacional de los 3 entornos, medido por el tiempo de cálculo necesario por hora (Valor más bajo mejor) . . . .	117
7.4.1	Resultado de los algoritmos de referencia del entorno 3 . . . .	122

7.4.2	Resultado de los algoritmos de referencia del entorno 3 . . . . .	123
7.4.3	Ventajas y desventajas de los algoritmos de comparación . . . . .	125
8.1.1	Intersección de dos corredores bidireccionales con caminos múltiples . . . . .	129
8.1.2	Estado inicial de la intersección del caso de uso de la FASE I . . . . .	130
8.1.3	Estado inicial de la intersección del caso de uso de la FASE I en detalle . . . . .	131
8.3.1	Resultado del desempeño con diferentes metodologías de selección en los algoritmos genéticos (Valor más alto mejor) . . . . .	136
8.3.2	Diagrama del aprendizaje de <i>GA1</i> . . . . .	140
8.3.3	Ejemplo de una tabla de segmentos . . . . .	142
8.3.5	Ejemplo de una tabla de colisiones de la intersección de la Figura 8.3.4 . . . . .	147
8.3.4	Ejemplo visual de la representación de una intersección . . . . .	147
8.3.6	Diagrama del algoritmo de la Fase I . . . . .	149
8.4.1	Resultados del número máximo de AGV que pasan por la intersección de la Fase I . . . . .	151
8.4.2	Número de vehículos que atraviesan una intersección durante el aprendizaje de <i>GA1</i> (Valor más bajo mejor) . . . . .	153
9.3.1	Diagrama del aprendizaje de <i>GA2</i> . . . . .	162
9.3.2	Diagrama del algoritmo <b>GA2</b> de la Fase II . . . . .	165
9.4.1	Resultados de la Fase II en el entorno 3 . . . . .	168
9.4.2	Carga computacional de la Fase II en el entorno 3, medido por el tiempo de cálculo necesario por hora (Valor más bajo mejor) . . . . .	169
9.4.3	Resultados durante el aprendizaje de <i>GA2</i> (Valor más bajo mejor) . . . . .	170
10.1.1	Ejemplo de la representación del <i>Roadmap</i> en el algoritmo de la Fase III . . . . .	175
10.2.1	Diagrama del cálculo de la posición futura de los AGV . . . . .	181
10.2.2	Diagrama del algoritmo del movimiento dinámico de los AGV . . . . .	183
10.2.3	Diagrama del algoritmo de selección de zonas . . . . .	185
10.2.4	Ejemplo de la secuencia de activación de una Red de Petri . . . . .	195
10.2.5	Diagrama de representación de si un punto se encuentra dentro de un rectángulo . . . . .	198
10.2.6	Representación de los puntos en un rectángulo . . . . .	200
10.2.7	Diagrama del cálculo de la detección generada por el tráfico . . . . .	201
10.3.1	Diagrama del cálculo de la ruta de los AGV . . . . .	204
10.3.2	Diagrama ejemplo de un grafo del entorno para el cálculo de la ruta de los AGV . . . . .	205
10.3.3	Diagrama del algoritmo de resolución del grafo de la ruta . . . . .	206
10.3.4	Diagrama ejemplo de la resolución del grafo de la ruta . . . . .	208
10.3.5	Diagrama ejemplo de la resolución del grafo de la ruta . . . . .	216
10.5.1	Diagrama del algoritmo de aprendizaje de la Fase III . . . . .	224

10.5.2	Comparativa de los resultados con las diferentes propuestas de información a utilizar por una intersección de las demás intersecciones (Valor más alto mejor) . . . . .	227
10.5.3	Diagrama del algoritmo de la gestión de las intersecciones del algoritmo <i>GA3</i> . . . . .	233
10.6.1	Diagrama del proceso de generación de un gestor de flotas adaptado al entorno por parte del algoritmo <i>GA3</i> . . . . .	236
10.6.2	Funcionamiento global del algoritmo <i>GA3</i> . . . . .	238
11.1.1	Resultados de la Fase III en el entorno 1 . . . . .	242
11.1.2	Resultados de la Fase III en el entorno 2 . . . . .	244
11.1.3	Resultados de la Fase III en el entorno 3 . . . . .	245
11.1.4	Carga computacional de los diferentes entornos en la Fase III (Valor más bajo mejor) . . . . .	247
11.1.5	Resultados durante el aprendizaje de <i>GA2</i> y <i>GA3</i> en el Entorno 3, con valor de <i>PST</i> del 80 % (Valor más bajo mejor) . . . . .	249
11.1.6	Tiempo de aprendizaje de <i>GA2</i> y <i>GA3</i> (Valor más bajo mejor) . . . . .	250
11.2.1	Resultados globales absolutos del entorno real . . . . .	251
11.2.2	Resultados globales respecto al <i>STPRAN</i> del entorno real . . . . .	251
11.2.3	Resultados de tiempo de viaje absolutos en el entorno real . . . . .	253
11.2.4	Resultados de tiempo de viaje respecto al <i>STPRAN</i> en el entorno real . . . . .	253
11.2.5	Visualización del movimiento de los AGV durante la ejecución de las tareas T3_1-3 y T3_3-1, utilizando el algoritmo <i>STPRAN</i> en el entorno real . . . . .	254
11.2.6	Visualización del movimiento de los AGV durante la ejecución de las tareas T3_1-3 y T3_3-1, utilizando los algoritmos <i>STPCT</i> o <i>GA3</i> en el entorno real . . . . .	255
11.2.7	Visualización del movimiento de los AGV durante la ejecución de las tareas T4_3-2 y T4_1-3, utilizando los algoritmos <i>STPCT</i> o <i>STPRAN</i> en el entorno real . . . . .	256
11.2.8	Visualización del movimiento de los AGV durante la ejecución de las tareas T4_3-2 y T4_1-3, utilizando el algoritmo <i>GA3</i> en el entorno real . . . . .	256
11.2.9	Visualización del movimiento de los AGV durante la ejecución de las tareas T9_3-2 y T9_2-1, utilizando los algoritmos <i>STPRAN</i> o <i>STPCT</i> en el entorno real . . . . .	257
11.2.10	Visualización del movimiento de los AGV durante la ejecución de las tareas T9_3-2 y T9_2-1, utilizando el algoritmo <i>GA3</i> en el entorno real . . . . .	257
11.3.1	Rendimiento de todos los algoritmos en los entornos en relación al <i>STPI</i> . . . . .	258
11.3.2	Rendimiento medio y desviación estándar de todos los algoritmos en los entornos en relación al <i>STPI</i> . . . . .	260

# Capítulo 1

## Motivación de la tesis, objetivos y estructura de la memoria

### Resumen

En este capítulo se expondrán la motivación, los objetivos y las contribuciones de la tesis. Se finalizará con la estructura de la memoria.

## 1.1 Motivación

El desarrollo económico está influenciado por diversos factores y descubrimientos, entre los cuales se encuentra el transporte, el cual ha permitido nuevas formas de movilización de recursos, dando lugar a nuevos paradigmas. Es importante destacar que el transporte ha sido una herramienta crucial para la expansión del comercio y el intercambio de bienes y servicios, lo que ha contribuido al crecimiento económico de las naciones. En este sentido, el transporte se ha convertido en un elemento clave en la economía global, y su evolución y mejora continúan siendo un factor determinante en el progreso económico de las sociedades. Hasta hace poco, el progreso en el transporte se centraba en el desarrollo de nuevos medios de transporte cada vez más eficientes y de mayor rendimiento, siempre pilotados por seres humanos. Sin embargo, en la actualidad, las nuevas tecnologías están transformando el transporte mediante la introducción de vehículos autónomos. Un ejemplo de este cambio se puede ver en el sector del e-commerce, donde empresas como Amazon han revolucionado la forma en que se almacenan los productos, al cambiar la recogida manual por la recogida automatizada mediante vehículos autónomos. Este avance tecnológico no solo mejora la eficiencia de las operaciones, sino que también plantea nuevas posibilidades.

La introducción de los vehículos autónomos ha demostrado ser una innovación en términos de eficiencia y productividad. Sin embargo, es importante destacar que también se presentan nuevas problemáticas que requieren ser resueltas. Una de las principales cuestiones a abordar es la gestión coordinada de múltiples vehículos autónomos, dado que su funcionamiento prescinde de la intervención humana en el pilotaje, lo que implica la necesidad de sistemas que reemplacen esta función.

Para abordar la cuestión de la coordinación de vehículos autónomos, es común el empleo de un sistema conocido como Gestor de flotas. La implementación de un Gestor de flotas resulta crucial para maximizar las ventajas que ofrecen los vehículos autónomos, ya que permite mejorar la eficiencia y productividad de las operaciones, reducir costos y minimizar el impacto ambiental. De esta manera, la Gestión de flotas se convierte en un elemento indispensable al momento de desplegar vehículos autónomos. Es importante destacar que el uso de un Gestor de flotas contribuye a la optimización del rendimiento de los vehículos autónomos, al permitir la coordinación y el control de múltiples unidades de forma eficiente y segura. En definitiva, la implementación de un Gestor de flotas es una estrategia clave para maximizar los beneficios de los vehículos autónomos y garantizar su correcto funcionamiento.

El Gestor de flotas es un algoritmo que se encarga de la administración

de un grupo de vehículos de transporte que han sido reunidos bajo un mismo propósito por una organización. Sus funciones abarcan distintas tareas, como ocuparse del mantenimiento de las unidades de la flota, administrar los sistemas de seguimiento, planificar los recorridos y llevar un control riguroso de los costes asociados a cada decisión en la gestión, entre otras. A través de estas acciones, la Gestión de flotas rentabiliza la inversión en vehículos y equipamiento para la logística. Su objetivo fundamental es optimizar la eficiencia y productividad de una flota en todas las etapas de su operación.

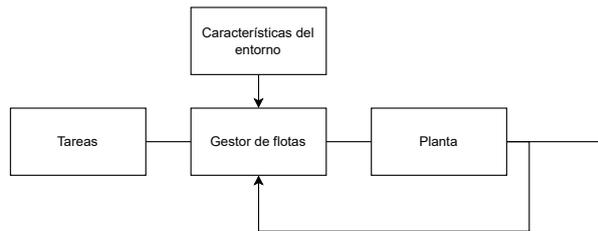
Por tanto, aunque la introducción de vehículos autónomos plantea nuevas problemáticas, la implementación de soluciones tecnológicas como los Gestor de flotas permiten superar estos desafíos y aprovechar al máximo los beneficios que ofrecen los vehículos autónomos.

Existen diferentes factores que pueden alterar la eficacia de un Gestor de flotas dependiendo del lugar de actuación, estos factores son los siguientes:

1. **El espacio de las instalaciones.** Este determina, en gran medida, el rendimiento del Gestor de flotas, pudiendo ser el entorno el que esté adaptado a las necesidades del Gestor de flotas o, por el contrario, teniendo el Gestor de flotas que adaptarse al entorno.
2. **El medio.** Los medios de transporte que se analizarán en esta tesis son los vehículos autónomos, los cuales presentan una amplia variedad de tamaños, morfologías, funciones y cargas que transportar, dependiendo del trabajo que deban realizar. En entornos intralogísticos es común el uso de los Vehículos Guiados Automáticamente (AGV, por sus siglas en inglés, Automated Guided Vehicle).
3. **El trabajo a realizar.** Están determinadas por las tareas que tienen que realizar los vehículos autónomos, siendo específicas de las necesidades del entorno.

Cualquier variación en los factores mencionados puede influir en el método de actuación del Gestor de flotas. Como resultado, la eficacia de los algoritmos utilizados para coordinar estos vehículos se ve afectada por el entorno de operación. Por esta razón, cada algoritmo de Gestión de flotas se diseña considerando un tipo de entorno. No obstante, estos algoritmos no están diseñados para cada entorno específico, lo que indica un potencial de mejora. Este potencial de mejora es el enfoque principal de esta tesis y se aborda mediante **la hipótesis de que la generación de algoritmos Gestión de flotas específicamente adaptados para cada entorno permitiría una toma de decisiones específica para el entorno, mejorando así su rendimiento global.**

Para ilustrar el cambio en la metodología de la Gestión de flotas propuesto en esta tesis, primero es necesario presentar la estructura de control convencional en la Gestión de flotas. De esta manera, se podrán describir los pasos necesarios para implementar un Gestor de flotas en una instalación y, en consecuencia, destacar las modificaciones en el enfoque sugerido en esta tesis.



*Figura 1.1.1:* Estructura típica de control de un Gestor de flotas

Una manera de representar el sistema de control típico en Gestión de flotas, es la que se muestra en la Figura 1.1.1, si nos centramos en Gestor de flotas, este cuenta con las siguientes categorías de entradas y salidas de control:

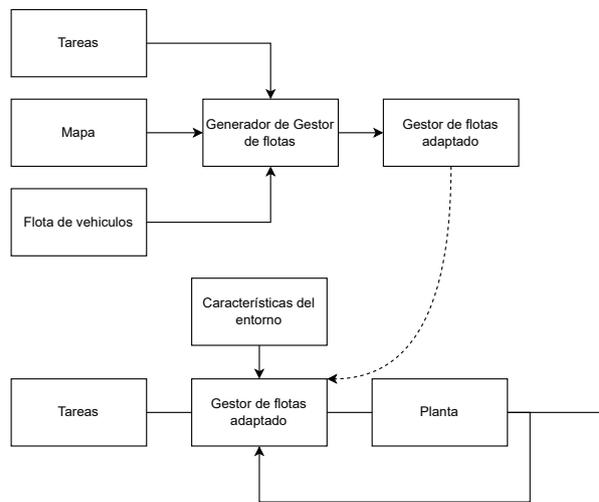
- **Datos de entrada:** Los datos procesados por el Gestor de flotas se dividen en datos estáticos, conocidos previamente, y datos dinámicos:
  - **Datos estáticos:** Estos son los datos conocidos del entorno nombrados como características del entorno, formados por los puntos de recarga, el mapa, entre otros aspectos. Esto se utiliza principalmente para que los algoritmos de generación de rutas de los Gestor de flotas puedan determinar las rutas de los vehículos.
  - **Datos dinámicos:**
    - \* **Tareas:** Estas incluyen todas las solicitudes de trabajo enviadas al Gestor de flotas, las cuales constituyen el trabajo que tiene que realizar la flota de vehículos.
    - \* **Retroalimentación del entorno:** Todos los datos sobre el estado actual de la planta, destacando principalmente el movimiento y las acciones llevadas a cabo por la flota de vehículos.
- **Datos de salida:** Estos se refieren a las instrucciones proporcionadas a la planta, como las rutas que seguirá la flota de vehículos y las acciones a ejecutar, entre otras cosas.

Así, el control llevado a cabo por el Gestor de flotas implica enviar órdenes a los vehículos para que puedan completar el trabajo, considerando el estado de la planta y sus características.

Habiendo expuesto el sistema de control empleado por los Gestores de flotas, podemos clasificar las distintas fases necesarias para implementar un Gestor de flotas en los siguientes pasos:

1. **Diseño del Gestor de flotas:** En esta etapa, se implementan los algoritmos de Gestión de flotas diseñados para la tipología del entorno objetivo.
2. **Instalación del Gestor de flotas:** Durante esta fase, se configuran los parámetros específicos de la instalación, siendo el mapa el parámetro más relevante. También se instala el Gestor de flotas en el hardware que lo ejecutará y se configuran las conexiones con la planta, incluyendo los vehículos.
3. **Funcionamiento del Gestor de flotas:** Por último, el algoritmo de Gestión de flotas lleva a cabo sus funciones en la instalación, siguiendo el método de control mostrado en la Figura 1.1.1.

Por lo tanto, aunque el Gestor de flotas se diseñe pensando en una tipología de entorno, el Gestor de flotas no está diseñado específicamente para el entorno, lo que deja una potencial de mejora por explotar.



**Figura 1.1.2:** Estructura propuesta de generación y control de un Gestor de flotas específica para un entorno

En esta tesis, se propone cambiar la metodología convencional y, en vez de desarrollar un algoritmo orientado a una tipología de entorno, sugerir un algoritmo generador de Gestor de flotas adaptado a la medida de cada entorno. Al adoptar este enfoque, las diferentes etapas requeridas para implementar el

Gestor de flotas serían las siguientes:

1. **Diseño del generador de Gestor de flotas:** En esta etapa, se desarrolla el algoritmo generador de Gestor de flotas.
2. **Generar el Gestor de flotas mediante el generador de Gestores de flotas:** Una vez obtenidos los parámetros de la planta, se emplea el generador de Gestores de flotas para crear un Gestor de flotas específicamente adaptado al entorno. Este proceso se representa en la parte superior de la Figura 1.1.2.
3. **Instalación del Gestor de flotas:** Se instala el Gestor de flotas generado específicamente para este entorno en el hardware que lo ejecutará y se configuran las conexiones con la planta, incluyendo los vehículos.
4. **Funcionamiento del Gestor de flotas:** Finalmente, el algoritmo generado se utiliza en la arquitectura convencional de Gestión de flotas para llevar a cabo las funciones habituales en una instalación, con la diferencia de que el algoritmo ha sido adaptado específicamente para esta instalación. Este aspecto se muestra en la parte inferior de la Figura 1.1.2.

En resumen, la metodología propuesta consiste en desarrollar un algoritmo capaz de generar Gestores de flotas adaptados a entornos específicos, en lugar de utilizar un único algoritmo de Gestión de flotas para todos los casos.

## 1.2 Objetivos y contribuciones

**El objetivo principal de esta tesis doctoral es la reducción de costes en el entorno intralogístico mediante el desarrollo de un algoritmo capaz de generar Gestores de flotas adaptados a entornos específicos.** Para lograrlo, se aplicarán técnicas de optimización con el fin de mejorar la eficiencia en la planificación y coordinación de las operaciones de la flota.

Para realizar el objetivo principal de esta tesis se proponen los siguientes subobjetivos:

1. **Definir el entorno:** En primer lugar, se presenta una descripción detallada y concreta del entorno intralogístico en el cual se desarrolla esta tesis. Se describen las características físicas del espacio y las instalaciones, así como las actividades y operaciones que se llevan a cabo en el mismo. Asimismo, se exponen las necesidades y requisitos específicos de la gestión de flotas de vehículos autónomos en este entorno, lo que permitirá comprender mejor los retos y desafíos que se plantean en el desarrollo de un algoritmo de Gestión de flotas eficiente y rentable. Concretar el entorno intralogístico es esencial para la comprensión del contexto en el que se desarrolla esta

investigación y para la identificación de las necesidades y oportunidades de mejora en la gestión de flotas de vehículos autónomos. El resultado de este subobjetivo es el siguiente:

- **Concretar el entorno intralogístico:** Especificación del entorno intralogístico que abarca esta tesis en el ámbito de la Gestión de flotas.
2. **Sistemas de medición de rendimiento:** Una vez definidas las condiciones de trabajo del Gestor de flotas, se procederá a desarrollar una metodología de cálculo del rendimiento orientado a costes, en función de los criterios y restricciones establecidos en el apartado 1. Posteriormente, un simulador explícitamente diseñado permitirá obtener el rendimiento de una flota de vehículos autónomos en un entorno específico y el rendimiento máximo teórico posible. El resultado de este subobjetivo es el siguiente:
    - **Simulador de flotas:** Un simulador capaz de extraer el rendimiento de Gestores de flotas de AGV en un entorno. Junto al rendimiento máximo ideal.
  3. **Estudio del desempeño de los Gestores de flotas de referencia:** Con el fin de determinar si el algoritmo propuesto en esta tesis es capaz de reducir los costes de la flota de vehículos autónomos en el entorno intralogísticos, se deberá conocer el rendimiento de los algoritmos actuales (Conocidos como algoritmos de referencia) en este tipo de entorno. Esto permitirá identificar los apartados en los que se puede obtener una mayor mejora con el nuevo algoritmo. El resultado de este subobjetivo es el siguiente:
    - **Algoritmos de referencia:** Un conjunto de algoritmos que serán utilizados para determinar que el algoritmo propuesto tiene un mayor rendimiento.
    - **Casos de uso:** Exposición de 3 casos de uso donde se evaluarán los Gestores de flotas. Adicionalmente, se expone un caso real en el que se realizarán las pruebas finales.
    - **Resultados de rendimiento de los algoritmos de referencia:** Evaluación de la reducción de costes de los algoritmos de referencia en los 3 casos de uso expuestos.
  4. **Propuesta de un algoritmo generador de Gestores de flotas adaptado específicamente a cada entorno:** Tras haber establecido los criterios de evaluación, la metodología para calcular el rendimiento máximo teórico y el rendimiento de los algoritmos de referencia, se propone desarrollar un nuevo algoritmo generador de Gestores de flotas que se adapten a cada entorno específico y que permita reducir los costes en comparación con los algoritmos de referencia en los entornos analizados. El resultado obtenido a partir de este subobjetivo es el siguiente:
    - **Algoritmo generador de Gestores de flotas:** El nuevo algoritmo propuesto será capaz de generar Gestores de flotas adaptados específicamente a cada entorno mediante aprendizaje y obtener mejores resultados que los algoritmos de referencia en los entornos descritos.
  5. **Validación de hipótesis:** Finalmente, se procederá a la evaluación del algoritmo propuesto, con el objetivo de verificar que se cumple el objetivo principal de esta tesis doctoral: la reducción de costes mediante un Gestor

de flotas en entornos intralogísticos con vehículos autónomos. El resultado de este subobjetivo es el siguiente:

- **Validación de la hipótesis y del objetivo:** Se realizará una comparativa de los resultados obtenidos por el nuevo algoritmo propuesto y los algoritmos de referencia en diferentes entornos para validar la hipótesis y el objetivo de reducción de costes mediante un Gestor de flotas en entornos intralogísticos con vehículos autónomos.

### 1.3 Estructura de la memoria

La presente tesis se divide en seis subobjetivos, cada uno de ellos conformado por uno o varios capítulos. El primer subobjetivo corresponde a la introducción de la tesis, mientras que los cinco subobjetivos siguientes son necesarios para completar el objetivo principal de la misma. En la Figura 1.3.1, se presenta un esquema que ilustra la relación entre los subobjetivos, capítulos, resultados y el objetivo principal de la tesis.

A continuación, se presenta un resumen de la estructura de cada subobjetivo en relación con los capítulos correspondientes de la tesis:

0. **Introducción:** En este primer apartado se realiza una pequeña introducción a la tesis mediante el siguiente capítulo:
  - **Capítulo 1. Motivación de la tesis, objetivos y estructura de la memoria:** En este capítulo se expondrán la motivación, los objetivos y las contribuciones de la tesis. Se finalizará con la estructura de la memoria.
1. **Definir el entorno:** En este apartado se procede con los conceptos básicos de la Gestión de flotas, con el objetivo de exponer y concretarán las condiciones de trabajo del Gestor de flotas en el ámbito intralogístico de esta tesis. Mediante el siguiente capítulo:
  - **Capítulo 2. Sistemas de gestión de flotas en el entorno intralogístico:** En este capítulo se introduce el uso de la Gestión de flotas en la logística de transporte, que cuenta con multitud de campos de aplicación con requerimientos muy distintos. Se continúa con la definición del entorno, los criterios de evaluación de los algoritmos de Gestión de flotas para la reducción de costes y las restricciones propias del caso de aplicación de la Gestión de flotas de la tesis.
2. **Sistemas de medición de rendimiento:** En este apartado se exponen las metodologías de cálculo del rendimiento máximo y del rendimiento de un algoritmo en un entorno. El objetivo de este apartado es exponer las herramientas necesarias para poder evaluar diferentes algoritmos de Gestión de flotas en diferentes entornos y determinar cuál es el rendimiento máximo teórico posible. Este apartado está formado por los siguientes

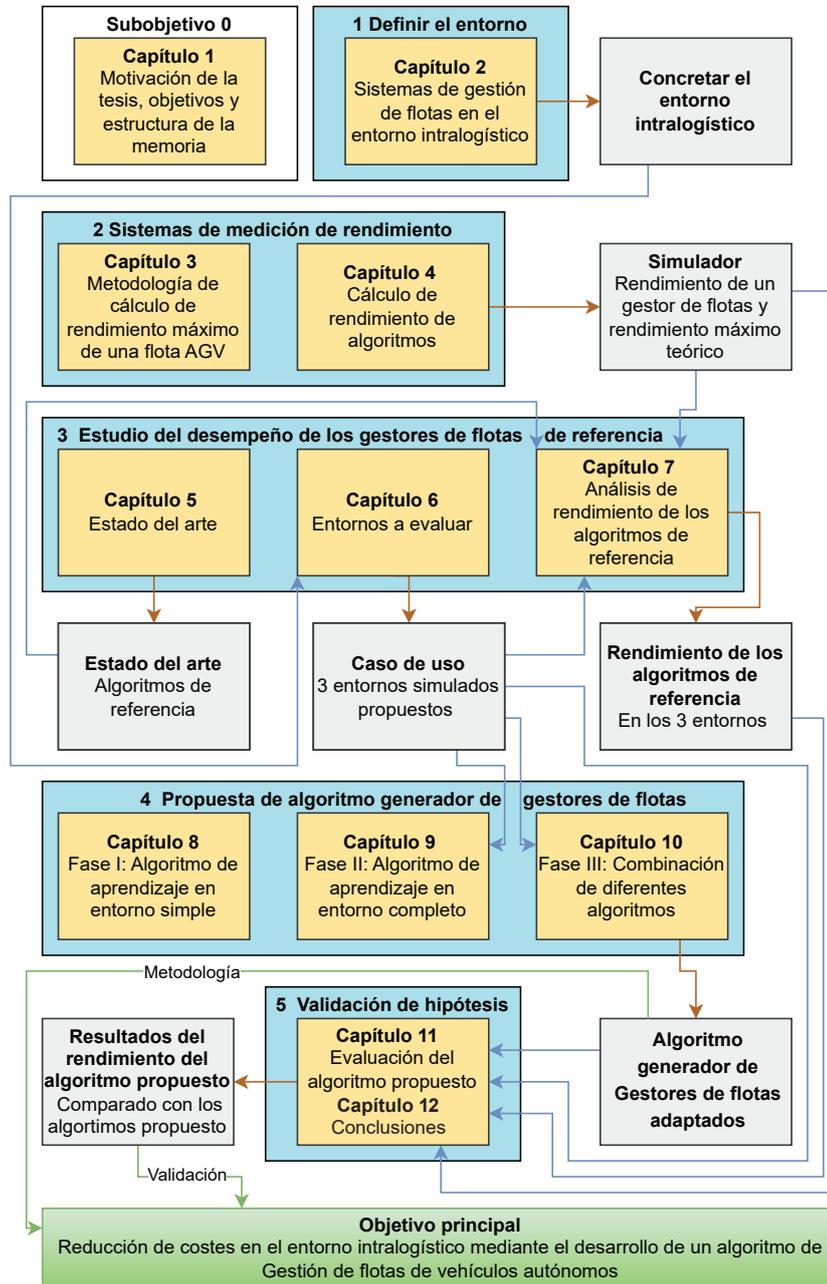


Figura 1.3.1: Estructura de la memoria en relación con los subobjetivos y objetivo de esta tesis

capítulos:

- **Capítulo 3. Metodología de cálculo de rendimiento máximo de una flota de AGV:** En este capítulo se expondrán diferentes metodologías de cálculo de rendimiento máximo de los AGV para las distribuciones de control de los AGV más comúnmente utilizados en la industria. Se expondrán las metodologías necesarias para extraer el rendimiento máximo teórico de un algoritmo de Gestión de flotas en un entorno específico.
  - **Capítulo 4. Cálculo de rendimiento de los Gestores de flotas mediante simulación:** En este capítulo se expondrá un software de simulación específico, desarrollado con el objetivo de poder evaluar el rendimiento de los Gestor de flotas en un entorno concreto, que incluye unos trabajos a realizar y unos vehículos específicos. Este simulador emula el comportamiento dinámico de los vehículos autónomos, y el Gestor de flotas va generando las consignas a los vehículos como si fueran vehículos reales. Los valores resultantes del uso de este simulador en un entorno y con un Gestor de flotas específicos pueden utilizarse para calcular la reducción de costes de la flota. También se puede extraer el rendimiento máximo teórico.
3. **Estudio del desempeño de los Gestores de flotas de referencia:** En este apartado se procederá con la evaluación del rendimiento en la reducción de costes de diferentes algoritmos de Gestión de flotas utilizados en la industria. Como primer paso, se expondrán diferentes algoritmos de Gestión de flotas. A continuación, se presentarán 3 casos de uso diferentes. Posteriormente, se llevará a cabo la evaluación del rendimiento de los algoritmos expuestos en los 3 casos de uso. Este apartado está compuesto por los siguientes capítulos:
- **Capítulo 5. Estado del arte:** En este capítulo se expondrán diferentes algoritmos de Gestión de flotas utilizados en la industria (nombrados en esta tesis como algoritmos de referencia). Además, se analizan varias metodologías de aprendizaje con el objetivo de identificar aquellas que podrían ser aplicables en el contexto de esta investigación.
  - **Capítulo 6 Entornos a evaluar:** En este capítulo se expondrán 3 casos de uso, incluyendo detalles sobre su espacio, medios y criterios de evaluación (Los parámetros que influyen en a reducción de costes). Estos entornos serán utilizados para evaluar los algoritmos del Capítulo 5 y también para la evaluación de los algoritmos propuestos en esta tesis. Adicionalmente, se expone un caso real en el que se realizaran las pruebas finales.
  - **Capítulo 7. Análisis de rendimiento de los algoritmos de referencia:** En este capítulo se presentarán los resultados de los algoritmos de referencia expuestos en el Capítulo 5, evaluados en los entornos presentados en el Capítulo 6. La evaluación constará de los siguientes apartados:
    - **Rendimiento de los algoritmos de Gestión de flotas en cada entorno:** En este apartado se cuantificará la reducción de costes

conseguida por cada algoritmo en cada uno de los entornos evaluados. Se compararán los resultados obtenidos por cada algoritmo para determinar cuál de ellos ofrece el mejor rendimiento en términos de reducción de costes.

- **Carga computacional de los algoritmos de Gestión de flotas en cada entorno:** En este apartado se analizará la carga de trabajo requerida por cada algoritmo en cada entorno, con el objetivo de evaluar su eficiencia en términos de recursos computacionales. Se medirá el tiempo de ejecución y la cantidad de recursos necesarios para cada algoritmo, con el fin de determinar cuál de ellos es más eficiente en términos de uso de recursos computacionales.
  - **Análisis de los algoritmos de Gestión de flotas:** En este apartado se realizará un análisis detallado del rendimiento de cada algoritmo, con el fin de identificar las condiciones de trabajo donde cada algoritmo presenta un mayor potencial de mejora. Se identificarán las fortalezas y debilidades de cada algoritmo para determinar en qué condiciones es más efectivo cada uno.
  - **Comparativa del rendimiento de los diferentes algoritmos de referencia:** En este apartado se comparará el rendimiento de los algoritmos de referencia entre sí, extrayendo las diferencias de rendimiento entre ellos en las diferentes situaciones analizadas.
  - **La mejor combinación:** En este apartado se identificarán las mejores características de cada algoritmo con el objetivo de utilizarlas en la creación de un nuevo algoritmo de Gestión de flotas. Se evaluarán las combinaciones posibles de características de los algoritmos existentes para crear un nuevo algoritmo que ofrezca un rendimiento superior en términos de reducción de costes y eficiencia computacional.
4. **Propuesta de un algoritmo generador de Gestores de flotas adaptado a distintos entornos:** En este capítulo se expondrán los algoritmos generadores de Gestión de flotas propuestos en esta tesis. La exposición se realizará mediante la propuesta de diferentes algoritmos de complejidad progresiva. Este apartado está formado por los siguientes capítulos:
- **Capítulo 8. Algoritmo generador de gestores de flotas: Fase I - Entorno simplificado:** En este capítulo se pretende demostrar la viabilidad de utilizar un algoritmo que se adapte al entorno mediante aprendizaje, dependiendo de las peculiaridades de este y los criterios de evaluación. Por ello, se presentará una primera aproximación a un algoritmo generador de Gestores de flotas, nombrado como *GA1*, capaz de realizar un aprendizaje del entorno simplificado.
  - **Capítulo 9. Algoritmo generador de gestores de flotas: Fase II - Entorno completo:** Este capítulo presenta el algoritmo *GA2*, diseñado para generar un Gestores de flotas personalizado y adaptado a un entorno completo en concreto.
  - **Capítulo 10. Algoritmo generador de gestores de flotas: Fase III - Algoritmo completo:** Este capítulo presenta el algoritmo propuesto

en esta tesis, que consiste en un generador de Gestores de flotas que combina algoritmos con diversas capacidades de adaptación al entorno, integrando algoritmos de aprendizaje y otras metodologías complementarias.

5. **Validación de hipótesis:** En este apartado se llevará a cabo la evaluación del algoritmo propuesto, con el objetivo de verificar que se cumple el objetivo principal de esta tesis doctoral (la reducción de costes mediante un Gestor de flotas en entornos intralogísticos con vehículos autónomos) mediante una prueba de hipótesis. Este bloque está compuesto por los siguientes capítulos:

- **Capítulo 11. Evaluación del algoritmo propuesto:** En este capítulo se evaluará la versión más completa del algoritmo, el *GA3*. Se realizará primero la evaluación en los 3 entornos propuestos, comparando su resultado con los demás. Seguidamente, se expondrán los resultados en un entorno real. Continuando con un análisis del rendimiento respecto a los algoritmos de referencia. Se finalizará mediante una prueba de hipótesis.
- **Capítulo 12. Conclusiones:** En este capítulo se expondrán las conclusiones de esta tesis.

## Capítulo 2

# Sistemas de gestión de flotas en el entorno intralogístico

### Resumen

En este capítulo se introduce el uso de la Gestión de flotas en la logística de transporte, que cuenta con multitud de campos de aplicación con requerimientos muy distintos. Se continúa con la definición del entorno, las restricciones propias del caso de aplicación y los criterios de evaluación de los algoritmos de Gestión de flotas para la reducción de costes de la tesis.

Los apartados expuestos en este capítulo son los siguientes:

- **Introducción:** Se realiza una introducción de la Gestión de flotas, estableciendo el ámbito de trabajo de esta tesis.
- **Definición del entorno:** Se exponen los diferentes elementos que definen el entorno, como el espacio de las instalaciones, los vehículos autónomos y el trabajo a realizar, con el objetivo de concretar el ámbito de actuación de esta tesis.
- **Restricciones:** Se exponen las restricciones que el Gestor de flotas debe tener en cuenta a la hora de ejercer su función.
- **Criterios de minimización de costes:** Se presentan diferentes criterios para la evaluación de la reducción de costes generada por una flota de vehículos controlados por un Gestor de flotas.

## 2.1 Introducción

Actualmente, el transporte logístico de mercancías es de vital importancia para el desarrollo industrial, comercial y por ende, económico de los países. Desde la globalización, la producción y la comercialización de productos han incrementado la dependencia del transporte logístico para llevar a cabo estas operaciones.

El movimiento de mercancías se puede realizar mediante diferentes medios de transporte: aviones, barcos, camiones, automóviles, personas; siendo un método habitual para la realización del transporte logístico, la agrupación de los medios de transporte en flotas, también conocido como flotas de vehículos. Esta agrupación suele realizarse con medios de transporte similares.

El Gestor de flotas es el principal responsable de la gestión de flotas, encargado de la logística del transporte y de actuar según las directrices establecidas por las empresas. Dichas directrices pueden ser de naturaleza estática o dinámica, lo que implica la necesidad de sistemas predictivos para el cálculo de costes y la gestión eficiente. En la actualidad, el papel del Gestor de flotas ha adquirido mayor relevancia debido a la globalización y al surgimiento de nuevos métodos de comercialización, los cuales demandan un gran esfuerzo en el ámbito logístico.

La Gestión de flotas se refiere a la supervisión de un grupo de vehículos de transporte que han sido unidos con un propósito específico por una organización. Un Gestor de flotas de vehículos debe llevar a cabo diversas funciones para garantizar una operación eficiente, segura y productiva. Algunas de estas funciones pueden incluir:

- **Planificación y asignación de vehículos:** El Gestor de flotas debe establecer la cantidad de vehículos necesarios para satisfacer la demanda laboral y asignarlos a rutas y funciones específicas.
- **Supervisión del rendimiento de los vehículos:** Monitorear el rendimiento de los vehículos en términos de eficiencia energética, velocidad, seguridad y otros parámetros clave para garantizar una operación óptima.
- **Monitoreo y respuesta a incidentes:** Implementación de sistemas de monitoreo que permitan detectar y responder a incidentes que involucren a los vehículos o el trabajo que realizan. Estos sistemas deben ser capaces de generar acciones preventivas, correctivas o advertencias, según la gravedad del incidente, para garantizar la seguridad y el correcto funcionamiento de la flota.
- **Análisis de datos y generación de informes:** Recolectar y analizar datos de los vehículos para así poder identificar tendencias, áreas de mejora y oportunidades de optimización en la operación de la flota.

- **Capacitación y soporte al personal:** Proporcionar formación y apoyo al personal involucrado en la operación y supervisión de vehículos, incluidos conductores de respaldo, técnicos de mantenimiento y otros roles relevantes.
- **Integración y monitorización de la flota en sistemas superiores:** Asegurar que la información y el estado de la flota de vehículos sean accesibles y visibles para sistemas superiores, ya sean sistemas de gestión, o humanos encargados de supervisar la operación.

El principal objetivo de un Gestor de flotas es gestionar una flota de vehículos con el propósito de optimizar los costes logísticos. A continuación, se desarrollarán los ámbitos de la gestión de flotas a distintos niveles.

Las habilidades que un Gestor de flotas debe estar relacionadas con el contexto en el que se lleva a cabo el transporte, ya sea internacional, nacional, local o intralógico. A continuación, se presentan cada uno de estos ámbitos:

1. **Ámbito internacional:** Los gestores de transporte de ámbito internacional son flotas de gran capacidad de transporte, que se mueven en ámbitos internacionales y dotadas de gran flexibilidad. Su característica principal es la necesidad de lidiar con normativas internacionales y propias de cada país. Lo que tiene un fuerte impacto en la naturaleza del Gestor de flotas. Unos de los ejemplos paradigmáticos de este tipo de flotas los encontramos en la italiana MSC, la danesa Maersk, la francesa CMA CGM, la taiwanesa Evergreen y la china Cosco en el ámbito marítimo. En transporte aéreo encontramos empresas como UPS Airlines, FedEx y DHL entre otras.
2. **Ámbito nacional:** El siguiente nivel en la clasificación según el ámbito se encuentra en los sistemas de flotas de transporte a nivel nacional. A diferencia de las de nivel internacional, en este caso, el conjunto de leyes que rigen su desarrollo son más o menos estáticas (desde un punto regional pueden variar ligeramente), lo que simplifica el trabajo del Gestor de flotas en el aspecto de las leyes. Podemos encontrarnos con sistemas de transporte de ámbito público, como los sistemas de transporte ferroviario de mercancías o sistemas de distribución de bienes públicos, como Correos en España. La iniciativa privada, en general, desarrolla Gestor de flotas optimizados para la reducción de costes, lo que da lugar a empresas que ejercen las mismas funciones con costes menores a los que puede tener el estado.
3. **Ámbito local:** Si seguimos reduciendo el ámbito en el cual la flota desarrolla su trabajo nos encontramos con las flotas de carácter local (urbano, regional, etc.). La variedad de flotas que aparece en esta clase de ámbitos es extensa. Podemos encontrar flotas de transporte de bienes y/o de personas. Unos ejemplos serían los siguientes:
  - Los sistemas de transporte público, ponen en de relieve la impor-

tancia del Gestor de flotas. Podemos encontrar redes de transporte de personas con iniciativa pública (red de autobuses, metro, etc.), sistemas de transporte público con iniciativa privada (redes de taxis, sistemas como Uber y Cabify, etc.). Las diferencias entre ellas pueden desencadenar grandes saltos en los costes finales. Un ejemplo de esto lo encontramos en los últimos conflictos que se han dado entre los sistemas de gestión de taxis y las empresas de vehículos de transporte con conductor (VTC). La inclusión de las tecnologías de seguimiento y cálculo óptimo de rutas en las empresas VTC ha reducido los costes de gestión hasta el punto en el que los sistemas de coordinación de taxis han dejado de ser competitivos.

- Otros ejemplos paradigmáticos en este ámbito en los que se han producido mejoras sustanciales en el sistema de Gestión de flotas lo encontramos en empresas de reparto local como Glovo, Delivero o JustEat. Las mejoras incorporadas en los Gestión de flotas han permitido la aparición de múltiples empresas en esta área.
4. **Ámbito intralogístico:** El transporte puede producirse entre diferentes localizaciones de una nave industrial; esto impone restricciones particulares que no tienen otras flotas de vehículos y que requieren de cierto grado de especialización. A su vez, implican ciertas ventajas que pueden ser explotadas con el fin de reducir los costes de producción. En este ámbito intralogístico existen dos grandes casos de uso de transporte logístico, que proviene de la actividad que se realice en el emplazamiento, siendo estos grupos:
- Centros logísticos: Donde la flota de vehículos se encarga de la recepción de productos para llevarlos a los almacenes, y posteriormente llevar estos productos de los almacenes a los vehículos de transporte destinados a enviar estos productos. Un ejemplo de este tipo de empresas de logística serían las empresas de e-commerce, como puede ser Amazon o Alibaba.
  - Centros de producción: En este tipo de industria, el transporte de logística se centra en el suministro de elementos de la línea de montaje para el proceso productivo. Un ejemplo de sector industria sería el de automoción, contando con empresas de gran tamaño como puede ser el grupo Volkswagen o Toyota.

Esta tesis centrará su estudio en la Gestión de flotas del **ámbito intralogístico**.

Otro factor que afecta a la capacidad del Gestor de flotas es si están destinadas a la gestión de vehículos autónomos o de vehículos controlados manualmente. El caso de análisis de esta tesis será la gestión de **vehículos autónomos**.

La Gestión de flotas de vehículos autónomos depende en gran medida de la

convivencia de estos con otros elementos. Según la relación con dichos elementos, se pueden clasificar en:

1. **Vehículos en entornos aislados:** Estos vehículos se mueven independientemente de otros vehículos, personas u otros elementos, minimizando así las probabilidades de elementos externos que puedan perturbar la gestión. Por norma general, es habitual que el entorno se adapte a las necesidades del Gestor de flotas y la flota de vehículos. Un ejemplo paradigmático se puede encontrar en los sistemas de gestión de almacén de empresas como Amazon o Alibaba: grandes almacenes en los que sistemas automatizados se encargan de completar los pedidos de los usuarios sin intervención humana (o, al menos, con un mínimo de intervención).
2. **Vehículos en entornos compartidos:** Estos vehículos se mueven por los recintos industriales junto a otros vehículos (normalmente controlados manualmente), personas u otros elementos que pueden interponerse en el camino de los vehículos, generando incertidumbre en la gestión de los vehículos. Siendo en este caso el Gestor de flotas y la flota de vehículos, el que se adapta al entorno. Un ejemplo paradigmático se puede encontrar en las líneas de montaje de las cadenas de producción del grupo Volkswagen o Toyota: el movimiento de materiales o el suministro de material dentro de una línea de montaje secuencial, entre otros.

Esta tesis centrará su estudio en la Gestión de flotas de **vehículos en entornos compartidos**.

Por lo tanto, en esta tesis se centrará en la Gestión de flotas de **vehículos autónomos en entornos compartidos** dentro del **ámbito intralogístico**.

## 2.2 Definición del entorno

La aplicación de esta tesis se centra en la Gestión de flotas de **vehículos autónomos en entornos compartidos** dentro del **ámbito intralogístico**. En esta sección se pretende profundizar más en las características del entorno, mediante los siguientes apartados:

1. **El espacio de las instalaciones:** La morfología y particularidades del entorno donde se encuentran las instalaciones influyen significativamente en el rendimiento del Gestor de flotas, ya que deben adaptarse a estas condiciones.
2. **El medio:** Los vehículos autónomos son el medio de transporte analizado en esta tesis. Estos pueden variar en tamaño, forma, funciones y capacidad de carga, dependiendo del trabajo a realizar.

3. **El trabajo a realizar:** Las actividades que deben llevar a cabo los vehículos autónomos están determinadas por las necesidades específicas del entorno en el que operan.

### 2.2.1 El espacio de las instalaciones

El espacio de las instalaciones es un factor crucial en la gestión de flotas de **vehículos autónomos** que operan en **entornos compartidos** dentro del **ámbito intralogístico**. Estas instalaciones pueden incluir almacenes, centros de distribución, fábricas o cualquier otro entorno donde los vehículos autónomos interactúan con personas, otros vehículos o infraestructuras.

La morfología y las particularidades de las instalaciones influyen en la eficiencia y efectividad del Gestor de flotas. Por ejemplo, la disposición y diseño de las instalaciones pueden afectar la accesibilidad y la capacidad de los vehículos autónomos para navegar y realizar sus actividades de manera eficiente. Además, las instalaciones pueden presentar obstáculos y desafíos específicos que deben ser abordados por el Gestor de flotas y los vehículos autónomos, como áreas restringidas, zonas de tráfico mixto, puntos de congestión, entre otros.

En contextos intralogísticos, si nos centramos en el los entornos suelen ser muy variados, sobre todo dependiendo del trabajo a realizar. Los entornos más comunes son los siguientes:

1. **Almacenes automatizados.** En estos emplazamientos, los vehículos circulan en un entorno destinado para su circulación. Por lo general, en dichos entornos, los vehículos pueden desplazarse únicamente a través de cuadrículas con distancias predeterminadas, lo que limita los movimientos que pueden realizar y les exige gestionar la posición de almacenamiento del material. Normalmente, estos almacenes suelen ser **entornos aislados**, pero la inclusión de nuevos elementos de seguridad permite, en algunos casos, que se instalen almacenes automatizados en **entornos compartidos**.
2. **Movimiento de material por el interior de fábricas.** En este caso, a diferencia del anterior, los vehículos tienen que adaptar sus recorridos al entorno. Este entorno puede ser muy variado, dependiendo de la distribución y el tamaño de la instalación. La coexistencia con otros elementos suele ser elevada. Este hecho afecta en gran medida a los Gestores de flotas.
3. **Líneas de montaje.** En la industria, se utilizan cada vez más los vehículos para sustituir a las líneas de montaje tradicionales. Para lograr que los vehículos puedan trabajar en este entorno, el Gestor de flotas debe enfrentarse a muchos retos, sobre todo en cuanto a las necesidades del entorno y al cumplimiento de normativas, pero, en el caso del Gestor de flotas, la capacidad de decisión es baja y los requerimientos que tienen que cumplir son, sobre todo, del ámbito de la ejecución, como pueden ser



*Figura 2.2.1:* Ejemplo de un AGV

sincronismos de velocidad, evitar colisiones y atrapamientos, etc.

Esta tesis se centrará en el caso de transporte logístico dentro del **Movimiento de material por el interior de fábricas**, incluyendo los casos en los que se circule por **Almacenes automatizados en entornos compartidos**. Estos entornos se caracterizan por ser los espacios más complicados en cuanto al entorno y la mayor cantidad de coexistencia con otros elementos.

## 2.2.2 Medios

En apartados anteriores se ha indicado que el tipo de medio de transporte para esta tesis sería vehículos autónomos, en entorno intralogísticos. Concretamente en el interior de las fábricas. Esta sección se centrará en la exposición de los medios de transporte, como son los vehículos autónomos.

Dentro del este ámbito, existe diferentes tipologías de vehículos autónomos, estos están determinados por el tipo de libertad que tiene por el sistema de navegación que utilizan, principalmente en la industria existen dos grandes tipos de clasificaciones:

- **Navegación mediante grafo.** El entorno se divide en un grafo donde hay nodos conectados por segmentos. Los vehículos pueden circular de un nodo al siguiente si existe un segmento que los conecte. Los vehículos solo pueden circular por encima de los segmentos. Esta característica define por dónde puede circular los vehículos, provocando que el sistema sea menos flexible para añadir nuevos destinos. Su ventaja reside en su comportamiento, es más determinista.
- **Navegación libre.** Con este tipo de navegación, los vehículos deciden por dónde circular. En algunos casos, se marcan las zonas por donde un vehículo no puede circular, pero, para el resto, es el propio vehículo el que decide la ruta. Pudiendo variar su navegación con mayor libertad. Su

desventaja es que su comportamiento, es menos determinista.

En la industria, se clasifican los vehículos autónomos dependiendo del tipo de sistema de navegación que utilicen, siendo los tipos más comunes los siguientes:

- **AGV** (Automatic Guided Vehicle): Su sistema de navegación está basado en **Navegación mediante grafo**. Siendo ideal para entornos donde la flexibilidad requerida no sea muy alta, pero se requiera un comportamiento determinado y constante. Un ejemplo donde se suelen utilizar estos tipos de vehículos es para el suministro de transporte logístico, a una línea de producción.
- **AMR**: (Autonomous Mobile Robot): Su sistema de navegación está basado en **Navegación libre**. Esto le otorga una mayor flexibilidad a la hora de realizar su actividad, pero su comportamiento es menos predictivo. Un ejemplo donde se suelen utilizar estos tipos de vehículos sería en un aeropuerto, donde el entorno puede cambiar las rutas posibles sin preaviso, pero un retraso puntual no es un problema grave.

En el caso de estudio se centrará en el uso de los **AGV**, porque son los más comúnmente utilizados en este tipo de industria por su comportamiento más determinista y permiten mayor control por el Gestor de flotas. Un ejemplo de AGV se puede observar en la Figura 2.2.1.

Los AGV por norma general son capaces de mover cargas mediante recogida y entrega automática de los elementos a transportar. En la Figura 2.2.2 se puede ver un AGV arrastrando un carro de carga en un entorno industrial. Este AGV se dirige a la línea de producción a depositar el material que carga; en el trayecto hasta su destino se encontrará con personas, con otros AGV, con bloqueos inesperados, etc. Que serán detectados por el vehículo, deteniéndose si fuera necesario.

Existen numerosos sistemas de navegación utilizados por los AGV. Principalmente están basados en **Navegación mediante grafo**. Dentro de esta tipología de sistemas de navegación, los más frecuentemente utilizados son: filoguiado óptico, filoguiado magnético, filoguiado virtual, navegación láser, GPS, visión estereoscópica, TOF, entre otros. La ventaja de que todos ellos fundamenten su tipología en el uso de un mapa de carreteras (*Roadmap*) mediante un grafo, minimiza el impacto del sistema de navegación de los AGV en el Gestor de flotas.

El Gestor de flotas debe contar con una descripción ajustada del compor-



*Figura 2.2.2:* Ejemplo de entornos industriales

tamiento dinámico de los AGV, si lo que se busca es optimizar los recursos. Las características dinámicas como la aceleración, la desaceleración o la velocidad máxima en rectas y en curvas pueden determinar de forma drástica la elección de un camino u otro al gestionar las flotas. Por ejemplo, un AGV con una menor capacidad de aceleración será penalizado al detenerse, por lo tanto, los caminos con más rectas y menos curvas pueden ser más adecuados aún siendo más largos.

### 2.2.3 El trabajo a realizar

En cada instalación hay un trabajo (una acción que realizar) o una función que deben ser realizados por los AGV; a partir de ahora nos referiremos a ellos como Tareas. Estas Tareas están formadas por una Tarea o más. El conjunto de Tareas dependen de las necesidades de cada instalación. Algunos ejemplos significativos de una Tarea pueden ser: mover el material desde un almacén a una línea de montaje, añadir un material dentro de un almacén o mover el producto en una línea de producción. Una forma de clasificar estas Tareas es basándonos en el tipo de movimiento requerido, pudiendo ser este un movimiento discreto o continuo:

- **Movimiento discreto.** La Tarea consiste en recoger un material en el destino  $A$  y llevarlo al destino  $B$ , pudiendo ser también una secuencia de destinos. En este tipo de Tarea, no existen restricciones específicas entre los diferentes destinos, lo cual proporciona flexibilidad al Gestor de flotas para decidir la ruta, detener los AGV y determinar la velocidad de cada

AGV. Este tipo de Tarea está asociado con la logística de transporte de materiales y puede tener restricciones de tiempo mínimo y máximo de entrega del material en cada destino.

- **Movimiento continuo.** La Tarea consiste en circular por un tramo donde la velocidad, la posición y las detenciones están condicionadas a las necesidades de la instalación. Por lo tanto, el Gestor de flotas debe asegurarse de que los AGV cumplan con las necesidades de la instalación. Este tipo de Tarea suele estar asociada a líneas de montaje.

En el ámbito de desarrollo del Gestor de flotas de esta tesis, las Tareas de las que nos ocuparemos serán las de **Movimiento discreto** de material en el interior de una instalación siguiendo las especificaciones de cada instalación.

Por consiguiente, en esta tesis se aborda el trabajo que deben realizar los AGV a través de la definición de Tareas. Cada Tarea implica que los AGV deben mover materiales para transportarlos hacia distintos destinos, ejecutando las acciones requeridas en cada uno de ellos. Es importante destacar que las Tareas pueden ser cíclicas y, en consecuencia, necesitar ser realizadas en múltiples ocasiones.

Para fragmentar una Tarea, cuando se produce una Petición en la instalación de una Tarea, se genera una Misión. Generando tantas misiones como peticiones de ejecución tenga una Tarea.

Los pasos a seguir para completar una Misión son los siguientes:

1. **Seleccionar un AGV:** El primer paso consiste en elegir el AGV más adecuado para llevar a cabo la Misión, considerando factores como el cumplimiento de los requisitos necesarios, su disponibilidad (suficiente nivel de batería, no estar asignado a otra Misión, etc.). A cada Misión solo puede asignarse un AGV y cada AGV solo puede tener asignada una Misión.
2. **Enviar al AGV al primer destino de la Tarea:** El AGV debe dirigirse al primer destino de la Tarea y realizar la acción correspondiente en dicha ubicación, como por ejemplo recoger o entregar objetos a transportar.
3. **Enviar al AGV de manera secuencial a todos los destinos de la Tarea:** El AGV debe desplazarse secuencialmente a todos los destinos de la Tarea y llevar a cabo la acción asignada en cada localización.
4. **Liberar el AGV y finalizar la Misión:** Una vez que el AGV haya alcanzado todos los destinos y completado las acciones de cada destino, se considera finalizada la Misión, liberando al AGV para que pueda realizar otras Misiones.

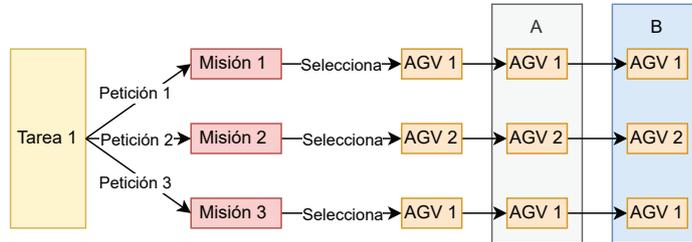


Figura 2.2.3: Ejemplo de ejecución de una Tarea y sus Misiones

En la Figura 2.2.3, se presenta un ejemplo ilustrativo de cómo se completa la Tarea 1, que consiste en recoger una mercancía en la posición A y dejarla en la posición B. La secuencia para completar la Tarea 1, para un total de 3 peticiones, es la siguiente:

1. **Generación de la petición 1:** Esta petición crea la Misión 1. Para completar la Misión 1 se requieren los pasos mencionados previamente, concretamente para esta Misión son los siguientes:
  - (a) **Seleccionar un AGV:** En este caso, se selecciona el AGV 1 porque cumple con las características necesarias para llevar a cabo la Misión y está disponible.
  - (b) **Enviar al AGV 1 al primer destino de la Tarea 1:** El AGV 1 se dirige al primer destino, en este caso es la posición A. Al llegar a la posición, realiza la acción, recoger la mercancía.
  - (c) **Enviar al AGV 1 al siguiente destino de la Tarea 1:** El AGV 1 se dirige al segundo destino, en este caso es la posición B. Al llegar a la posición, realiza la acción, depositar la mercancía.
  - (d) **Liberar el AGV 1 y finalizar la Misión 1:** Dado que se han completado todas las acciones de la Misión 1, se da por finalizada la Misión 1 y se libera el AGV 1.
2. **Generación de la petición 2:** Esta petición crea la Misión 2. Para completar la Misión 2 se requieren los pasos mencionados previamente, concretamente para esta Misión son los siguientes:
  - (a) **Seleccionar un AGV:** En este caso, se selecciona el AGV 2 porque cumple con las características necesarias para llevar a cabo la Misión y está disponible, a diferencia del AGV 1 que se encuentra realizando otra Misión.
  - (b) **Enviar al AGV 2 al primer destino de la Tarea 1:** El AGV 2 se dirige al primer destino, en este caso es la posición A. Al llegar a la posición, realiza la acción, recoger la mercancía.
  - (c) **Enviar al AGV 2 al siguiente destino de la Tarea 1:** El AGV 2 se dirige al segundo destino, en este caso es la posición B. Al llegar a la posición, realiza la acción, depositar la mercancía.

- (d) **Liberar el AGV 2 y finalizar la Misión 2:** Dado que se han completado todas las acciones de la Misión 2, se da por finalizada la Misión 2 y se libera el AGV 2.
3. **Generación de la petición 3:** Esta petición crea la Misión 3. Para completar la Misión 3 se requieren los pasos mencionados previamente, concretamente para esta Misión son los siguientes:
- (a) **Seleccionar un AGV:** En este caso, se selecciona el AGV 1 porque cumple con las características necesarias para llevar a cabo la Misión y está disponible, a diferencia del AGV 2 que se encuentra realizando otra Misión.
  - (b) **Enviar al AGV 1 al primer destino de la Tarea 1:** El AGV 1 se dirige al primer destino, en este caso es la posición *A*. Al llegar a la posición, realiza la acción, recoger la mercancía.
  - (c) **Enviar al AGV 1 al siguiente destino de la Tarea 1:** El AGV 1 se dirige al segundo destino, en este caso es la posición *B*. Al llegar a la posición, realiza la acción, depositar la mercancía.
  - (d) **Liberar el AGV 1 y finalizar la Misión 3:** Dado que se han completado todas las acciones de la Misión 3, se da por finalizada la Misión 3 y se libera el AGV 1.

## 2.3 Restricciones

La gestión de AGV realizada por el Gestor de flotas, al funcionar dentro de un contexto, está condicionada por el entorno. Si concretamos más, en el caso de esta tesis, se tratará el movimiento de material por el interior de las fábricas. El problema que deberá resolver el Gestor de flotas debe tener en cuenta diferentes restricciones. Si las clasificamos por su procedencia, se pueden dividir en los siguientes tipos:

- Restricciones de la industria
- Restricciones legislativas
- Restricciones del entorno
- Restricciones de los AGV

### 2.3.1 Restricciones de la industria

Cada industria tiene sus propias restricciones dependiendo del proceso que realiza y de las tareas de los AGV. En entornos intralogísticos, especialmente en industrias de fabricación, es habitual que los AGV tengan que entregar los elementos que transportan y que afectan directamente a la producción de la fábrica. La demora en la entrega de este material tiene un impacto directo en el proceso industrial, pudiendo causar problemas de funcionamiento en la fábrica. Por ello, es habitual que haya dos tiempos que cumplir por cada Tarea a realizar; uno es el tiempo objetivo *Tobj* de entrega, que se puede sobrepasar excepcionalmente;

y un segundo tiempo, llamado tiempo máximo de llegada permitido  $T_{max}$ , cuyo no cumplimiento implica la máxima penalización. La infracción de este segundo tiempo puede causar grandes inconvenientes a la industria donde se encuentran los AGV controlados por el Gestor de flotas.

### 2.3.2 Restricciones legislativas

Los AGV, al tener que trabajar en un entorno industrial, especialmente junto a personas, tienen que cumplir unas normativas reguladas por las instituciones gubernamentales. Dependiendo del lugar donde se pretendan instalar los AGV, estos tendrán que cumplir unas normativas particulares de cada región. Algunas de ellas son generalistas mientras que otras son específicas para AGV. Estas normativas tienen aspectos que pueden afectar al Gestor de flotas.

En la Unión Europea, la normativa específica que rige el uso AGV autónomos es la *ISO 3691-4* [19], que determina los requisitos que tienen que cumplir los AGV. Para simplificar su exposición, procederemos a exponer las peculiaridades que afectan al Gestor de flotas dividiendo la normativa en dos grandes grupos:

- **Restricciones de seguridad que limitan la movilidad del AGV.** Existen varias casuísticas que limitan la movilidad y la velocidad de los AGV. Estas pueden ser: la estabilidad de la carga, la navegación por entornos estrechos, la velocidad máxima del AGV en curvas, etc. Estos factores dan lugar a las siguientes restricciones:
  - Restricciones en el sentido de circulación. Normalmente dependen de la carga que lleven los AGV.
  - Velocidad máxima. Esta deberá ser reducida cuando haya ciertos peligros.
  - Velocidad máxima en curva. Según esta normativa, se limita la velocidad en las curvas, independientemente de que el AGV sea capaz de alcanzarla.
  - El AGV está obligado a indicar, mediante señales luminosas y/o acústicas, que va a empezar a moverse. Esta acción genera un retraso en el movimiento del AGV.
- **Restricción anticolidión.** Los AGV deben evitar colisionar con otros elementos. Para ello, cuentan con sensores perimetrales anticolidión con un nivel de seguridad determinado por la normativa. El objetivo de esta medida es detener el AGV antes de que colisione con personas, otros AGV u objetos. Estas detenciones anticolidión tienen que cumplir una categoría de seguridad que implica que el Gestor de flotas no pueda influir en estos aspectos, por lo que el AGV se detendrá sin posibilidad de que el Gestor de flotas pueda intervenir, una vez han sido activados.

### 2.3.3 Restricciones del entorno

Los entornos físicos de la industria donde se desarrollan este tipo de algoritmos generan sus propias restricciones. La primera afecta al *Roadmap*, ya que este suele ser irregular y normalmente está formado por muchos pasillos. Es habitual que venga definido por la instalación.

En este tipo de industrias, los AGV no son los únicos que circulan en exclusiva por el entorno, sino que existen elementos externos que conviven con los AGV, pudiendo detener los AGV si entran en su zona de detección perimetral. Los elementos más significativos que pueden convivir con los AGV son los siguientes:

- **Personas.** Estas causan detenciones de los AGV al convivir con ellos en las mismas zonas, activando temporalmente los elementos de seguridad anticolidión de los AGV y provocando detenciones, normalmente de corta duración.
- **Vehículos conducidos por personas.** Estos pueden ser Vehículo que circulan por el mismo entorno de manera excepcional (Vehículo de reparto de comida, de mantenimiento, de limpieza, movimiento de material no programado, etc.). Y, por otro lado, Vehículo que realizan tareas continuadas y similares a las de los AGV.
- **Otros vehículos autónomos controlados mediante otro gestor de tráfico.** Normalmente, en estos casos, se intenta la inclusión de los diferentes vehículos autónomos en el mismo Gestor de flotas.
- **Obstrucción de la zona de circulación.** Causada por elementos imprevistos que no están dentro de las condiciones normales de funcionamiento. Estas pueden ser elementos en medio de la calzada, incidentes, etc.

### 2.3.4 Restricciones de los AGV

Cada AGV tiene unas características que afectan a su movimiento, a las tareas que realizan, al tamaño... Todas ellas determinan el trabajo que pueden realizar, y cómo lo realizan. Estas características afectan directamente a las decisiones que tiene que tomar el Gestor de flotas. Las restricciones más significativas son las siguientes:

- **Velocidad máxima en recta**
- **Velocidad máxima en una curva dependiendo de sus características**
- **Aceleración y desaceleración**
- **Dimensiones del AGV**
- **Tipo de movilidad**
- **Las características del sistema de almacenaje de energía**

## 2.4 Criterios de minimización de costes

En el entorno expuesto, el Gestor de flotas tiene la responsabilidad de garantizar que todos los AGV completen todas las misiones correspondientes a cada tarea, cumpliendo con las restricciones del entorno y los requisitos de funcionamiento de los AGV, tales como la gestión energética, entre otros aspectos. Para lograrlo, el Gestor de flotas puede indicar a cada uno de los AGV la ruta que deben seguir y la velocidad objetivo, así como detener los AGV en cualquier momento y coordinar cuándo deben recargar sus baterías. El costo asociado dependerá de cómo los AGV lleven a cabo las tareas asignadas.

El problema a abordar es la minimización de los costes por parte de la instalación donde se pretenden utilizar estos AGV, pero siempre cumpliendo con los requisitos de dicha instalación. Existen dos factores que afectan en gran medida a los costes. Son los siguientes:

- **El número de AGV.** Al minimizar el número de AGV requeridos, se reduce el coste de adquisición del sistema global junto con el coste total de funcionamiento de los AGV.
- **Las misiones realizadas por los AGV.** Al maximizar las misiones que realizan los AGV, la instalación puede reducir sus gastos de funcionamiento.

Estos dos parámetros suelen ser contrapuestos, ya que si reducimos el número de AGV, esto podría conllevar una disminución en las misiones que son capaces de realizar. Por otro lado, si se requiere realizar más misiones, es probable que se necesite un mayor número de AGV. Dado que los costes asociados dependen de la instalación y de los AGV, proponemos los siguientes criterios de reducción de costes, denominadas *Cri1*, *Cri2*, *Cri3* y *Cri4*. La elección de la metodología adecuada dependerá del entorno específico.

Antes de presentar estas metodologías, se introducen diferentes indicadores que pueden ser utilizados en cada criterio para evaluar la reducción de costes. Los indicadores son los siguientes:

- **Número de misiones realizadas (*NMisCo*):** Determina directamente el trabajo realizado. Cuanto mayor sea este valor, mayor será el potencial de reducción de costes.
- **Número de AGV utilizados (*NAGV*):** Este indicador muestra cuántos AGV de los disponibles han sido utilizados, permitiendo conocer cuántos de los AGV son necesarios para cumplir las misiones. Cuanto menor sea este valor, mayor será el potencial de reducción de costes.
- **Tiempo medio de desplazamiento o viaje (*TMedV*):** Indica el tiempo

medio que los AGV tardan en realizar sus desplazamientos. Cuanto menor sea este valor, mayor será el potencial de reducción de costes.

- **Número de veces que los AGV han llegado a un destino después del  $T_{obj}$  requerido por la instalación ( $NT_{obj}$ ):** Este indicador determina la cantidad de veces que el AGV ha llegado más tarde del  $T_{obj}$ , lo cual podría generar incidencias en la instalación y, en consecuencia, un aumento de los costes. Cuanto menor sea este valor, mayor será el potencial de reducción de costes.
- **Número de veces que los AGV han llegado a un destino después del  $T_{max}$  requerido por la instalación ( $NT_{max}$ ):** Cuando un vehículo llega a algún destino después del  $T_{max}$ , se genera una incidencia en la instalación, lo que puede ocasionar pérdidas económicas debido a la interrupción de la actividad en la misma. Cuanto menor sea este valor, mayor será el potencial de reducción de costes
- **Tiempo medio de desplazamiento respecto al  $T_{obj}$  ( $TMedT_{obj}$ ):** Este indicador muestra el tiempo medio de llegada de los AGV en relación al tiempo  $T_{obj}$ . Cuanto menor sea este valor, mayor será el potencial de reducción de costes.
- **Tiempo medio de desplazamiento respecto al  $T_{max}$  ( $TMedT_{max}$ ):** Este indicador refleja el tiempo medio de llegada de los AGV en relación al tiempo  $T_{max}$ . Cuanto menor sea este valor, mayor será el potencial de reducción de costes.

Una vez expuestos los indicadores más relevantes, se procede a presentar los criterios de evaluación para la reducción de costes:

1. **Maximizar el trabajo que pueden realizar los AGV, manteniendo el espacio de la instalación constante, independientemente del número de AGV utilizado ( $Cr1$ ).**
  - La utilización de AGV en la industria se debe a que el coste de utilización de los AGV es inferior al de los métodos tradicionales, como vehículos conducidos manualmente o mediante tracción humana. Cuanto más trabajo se pueda realizar con los AGV, más se reducirán los costes. Asimismo, en menor medida, se tiene en cuenta el coste de utilizar un mayor número de AGV.
  - Los indicadores que determinan esta reducción, por orden de importancia, son los siguientes:
    - (a) **Número de misiones realizadas ( $NMisCo$ ):** Determina directamente el trabajo realizado.
    - (b) **Número de AGV utilizados ( $NAGV$ ):** Lo utilizaremos como criterio secundario cuando el  $NMisCo$  sea el mismo.
    - (c) **Tiempo medio de desplazamiento ( $TMedV$ ):** Este indicador nos permite determinar el rendimiento en aquellos casos en los que el resultado de los parámetros anteriores del algoritmo sean iguales.

Si los parámetros son diferentes, este criterio no será válido.

2. **Minimizar el número de AGV necesarios, manteniendo constante el espacio de la instalación y el trabajo que deben realizar los AGV (*Cri2*).**
  - Al minimizar el número de AGV requeridos para la misma cantidad de misiones, se produce una reducción directa de los costes de adquisición y del coste de funcionamiento de los AGV.
  - Las variables que determinan esta reducción, por orden de importancia, son las siguientes:
    - (a) **Número de AGV utilizados (*NAGV*):** Cuantos menos AGV se utilicen, más se reducirán los costes.
    - (b) **Número de veces que los AGV han llegado a cualquier destino después del *Tobj* marcado (*NTobj*):** Con este indicador, podemos distinguir el rendimiento para los casos en que el resultado del parámetro anterior del algoritmo sea el mismo.
    - (c) **Tiempo medio de desplazamiento (*TMedV*):** Con este indicador podemos evaluar las diferencias de rendimiento para los casos en los que el resultado del parámetro anterior (*NAGV*) sea el mismo.
3. **Maximizar el trabajo que pueden realizar los AGV, manteniendo constante el espacio de la instalación y el número de AGV utilizado (*Cri3*).**
  - En este caso, la reducción de costes se debe a que los AGV son capaces de realizar más misiones.
  - Las variables que determinan esta reducción, por orden de importancia, son las siguientes:
    - (a) **Número de misiones realizadas (*NMisCo*):** Determina directamente el trabajo realizado.
    - (b) **Número de veces que los AGV han llegado a cualquier destino después del *Tobj* marcado (*NTobj*):** Con este indicador, podemos distinguir el rendimiento para los casos en que el resultado del parámetro anterior del algoritmo sea el mismo.
    - (c) **Tiempo medio de desplazamiento respecto al *Tobj* (*TMedTobj*):** Con este indicador, podemos distinguir el rendimiento para los casos en que el resultado de los parámetros anteriores del algoritmo sea el mismo.
4. **Evaluación del desempeño para un número de AGV, un trabajo y un espacio de la instalación en concreto (*Cri4*).**
  - En este caso, la reducción de costes se evalúa directamente con el rendimiento de la flota de AGV ejecutando todas las misiones.
  - Las variables que determinan esta reducción, por orden de importancia, son las siguientes:
    - (a) **Número de misiones realizadas (*NMisCo*):** Con este valor se puede determinar si la flota de AGV ha sido capaz de completar todas las misiones.
    - (b) **Número de veces que los AGV han llegado a un destino después del *Tmax* (*NTmax*):** Con este indicador, podemos distinguir el rendimiento para los casos en que el resultado del parámetro

- anterior del algoritmo sea el mismo.
- (c) **Tiempo medio de desplazamiento o viaje** ( $TMedV$ ,  $TMaxV$  y  $TMinV$ ): Con esos indicadores, podemos distinguir el rendimiento para los casos en que el resultado del parámetro  $NMisCo$  y  $NT_{max}$  del algoritmo sea el mismo.
  - (d) **Tiempo medio de desplazamiento respecto al  $Tmax$**  ( $TMedT_{max}$ ,  $TMaxT_{max}$  y  $TMinT_{max}$ ): Este indicador tiene el mismo peso que el indicar anterior.

## Capítulo 3

# Metodología de cálculo de rendimiento máximo de una flota de AGV

### Resumen

Para calcular el rendimiento en términos de reducción de costes, es fundamental determinar el rendimiento máximo posible. En este capítulo, se presentan diversas metodologías para calcular el rendimiento máximo. Para ello, es necesario definir diferentes distribuciones de control de AGV, dado que estas influyen en el cálculo del rendimiento máximo.

Existen varios parámetros para cuantificar la reducción de costes, pero este capítulo se centra en el cálculo de la cantidad de AGV necesarios para llevar a cabo un número determinado de tareas en un espacio específico, considerando una distribución particular de AGV.

En primer lugar, se proponen diversas metodologías de cálculo analítico sin restricciones de tráfico, denominadas en esta tesis como *metodologías de cálculo analítico*. Se exponen fórmulas analíticas para las distintas distribuciones de control de AGV. No obstante, estas fórmulas de cálculo analítico parten de la premisa que el tiempo de petición de las tareas es constante, lo cual no siempre ocurre en la industria. Por este motivo, se propone una segunda metodología de

cálculo mediante programas genéricos de simulación, como el *software* Arena, denominada en esta tesis *metodologías de cálculo numérico*.

Las *metodologías de cálculo analítico* y *metodologías de cálculo numérico* se basan en que se conoce el tiempo de trayecto de los vehículos. Por esta razón, en este capítulo se presenta una metodología para calcular el tiempo de trayecto mínimo necesario para que un vehículo se desplace de un destino a otro en un entorno definido por un grafo, denominada como *metodologías de cálculo del tiempo de trayecto mínimo*.

Es importante tener en cuenta que las fórmulas anteriores no consideran el tráfico, lo cual es coherente, ya que el rendimiento máximo que se puede alcanzar cuando el tráfico no afecta a los vehículos. Sin embargo, existen otras restricciones, como el tiempo de operación requerido por los vehículos o las instalaciones y el tiempo de carga de las baterías de los vehículos, que también deben ser consideradas. Por esta razón, se continúa exponiendo la metodología de cálculo de rendimiento, pero incluyendo estas nuevas restricciones.

Por último, se presenta una metodología para calcular el rendimiento máximo en términos del número de vehículos necesarios para una distribución particular de AGV, un conjunto de tareas y un espacio específico, denominada como *metodologías de cálculo del tiempo de rendimiento máximo analítico*.

Se concluye que esta metodología es válida, pero requiere realizar acciones manuales para cada entorno. En el capítulo 4, se propone una metodología de cálculo automático mediante un simulador de AGV específicamente desarrollado para esta función.

Los apartados expuestos en este capítulo son los siguientes:

- **Distribución de los AGV:** En este apartado se describen las diferentes distribuciones de control de AGV más comúnmente utilizadas en la industria.
- **Cálculo de rendimientos sin restricción de tráfico:** Se presentan diversas metodologías de cálculo para determinar el número mínimo de AGV requeridos para realizar tareas específicas en un espacio determinado. Estos cálculos se llevan a cabo sin tener en cuenta las restricciones del tráfico, exponiendo las *metodologías de cálculo analítico* y *metodologías de cálculo numérico*.
- **Algoritmo de cálculo dinámico del tiempo de trayecto:** En este apartado se presenta una metodología para calcular el tiempo de trayecto requerido por un AGV para circular desde un destino a otro, sin considerar la restricción del tráfico, exponiendo la *metodología de cálculo del tiempo de trayecto*

*mínimo.*

- **Cálculos con restricción de tiempo de operación y carga de baterías:** Se presentan diferentes metodologías de cálculo para tener en cuenta las diferentes restricciones de tiempo de operación y carga de baterías de los AGV.
- **Extracción del rendimiento máximo mediante cálculo:** Se presentan diversas metodologías de cálculo del rendimiento máximo teórico de una flota de AGV en una instalación, exponiendo la *metodología de cálculo del tiempo de rendimiento máximo analítico*.

### 3.1 Distribución de los AGV

En el entorno industrial en el que planteamos el estudio de los AGV, se generan peticiones de tareas que deben ser realizadas por los AGV. Cada una de las peticiones genera una misión. El trabajo que debe realizar un AGV durante la misión está definido en la tarea, pero la selección de un AGV y el movimiento que tiene que hacer antes y después de la misión dependen de la arquitectura de distribución de los AGV. Esta asignación puede llevarse a cabo mediante diferentes metodologías. Dependiendo del entorno, la metodología puede estar determinada por este, o puede ser dejada a la libre elección del Gestor de flotas. A continuación, expondremos algunas de las metodologías que se utilizan habitualmente:

- **Distribución de los AGV con rutas cíclicas.** Cada AGV está asignado a una única tarea, lo que implica que, al llegar al destino final, se desplace de vuelta al destino inicial de esa misma tarea para esperar la siguiente misión de la misma tarea. En este tipo de rutas, se aprovechan los tiempos de proceso de la instalación para recargar baterías. La utilización de esta distribución está muy estandarizada, ya que son de fácil implementación y se pueden ejecutar mediante un sistema distribuido. Su eficiencia puede ser baja dependiendo de la relación entre la cantidad de tareas y la cadencia de las mismas.
- **Distribución de los AGV en rutas con punto de partida único.** Si todos los AGV parten del mismo emplazamiento, se puede asignar al primero de ellos la siguiente misión. Al terminar la misión, se dirige al punto inicial. Normalmente, se aprovecha este emplazamiento para realizar la carga de las baterías.
- **Distribución de los AGV con múltiples puntos de descanso para los AGV.** Los AGV cuentan con múltiples lugares en los que pueden esperar cuando no tienen misiones. En cada punto de espera, se decide la siguiente misión del primer AGV que esté en la cola de cada uno de estos puntos. La carga de batería se suele realizar en zonas específicas para ello.
- **Distribución libre de los AGV.** En esta distribución, cuando un AGV termina una tarea, se le asigna una nueva misión que suponga un mayor rendimiento, siendo enviado a zonas de descanso si no hay misiones que le puedan ser asignadas. La carga de batería se suele realizar en zonas específicas para ello. Esta es la distribución que mayor rendimiento puede obtener. Pero, a su vez, es la que requiere un mayor coste en integración y mantenimiento.

Normalmente, la distribución escogida para los AGV viene marcada por los requisitos de la instalación donde se pretenden instalar los AGV y el Gestor de flotas.

### 3.2 Cálculo de rendimientos sin restricción de tráfico

Para calcular el rendimiento, primero tenemos que concretar que valores serán los utilizados para cuantificar el rendimiento. En este apartado, proponemos como valor de rendimiento el número de AGV utilizados, para un espacio y unas tareas específicas.

Para que un AGV realice misiones, dependiendo de la distribución del entorno, puede tener que desplazarse a los siguientes emplazamientos:

- $A_n$ : Nodo que determina el destino inicial de una misión de la tarea  $n$ .
- $B_n$ : Nodo que determina el destino final de una misión de la tarea  $n$ .
- $C$ : Nodo que determina el punto de salida único.
- $D_m$ : Nodo que determina el punto de salida múltiple  $m$ .
- $D_{m'}$ : Nodo que determina el punto de salida múltiple  $m'$ .

En esta metodología se tiene en cuenta que las tareas solo cuentan con dos destinos, el  $A_n$  y  $B_n$ .

A continuación, procederemos a exponer la metodología de extracción del número mínimo de AGV necesarios para cada una de las tipologías de distribución de los AGV. Se presentarán las *metodologías de cálculo analítico* y las *metodologías de cálculo numérico*.

#### 3.2.1 Metodologías de cálculo analítico

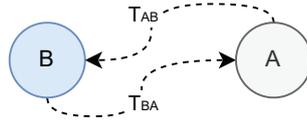


Figura 3.2.1: Esquema básico de la realización de una tarea

En este apartado se mostrarán las diferentes metodologías nombradas como *metodologías de cálculo analítico*. Una tarea consiste en llevar o mover material a una secuencia de destinos. Si simplificamos la tarea con dos destinos: el emplazamiento inicial  $A$  y el emplazamiento final  $B$ , al contar con una única tarea, los AGV recorren la ruta indicada en la Figura 3.2.1. Realizando el movimiento cíclico de  $A$ ,  $B$  y  $A$ , el número de AGV necesarios para completar esta tarea con una periodicidad  $Te$  se lleva a cabo mediante la siguiente fórmula:

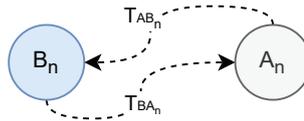
$$N_{AGV} = \left\lceil \frac{T_{AB} + T_{BA}}{T_e} \right\rceil \quad (3.2.1)$$

Donde:

- $N_{AGV}$ : Número de AGV necesarios para cumplir las tareas.
- $T_e$ : Periodo entre peticiones de una tarea. Medido en segundos.
- $T_{AB}$ : Tiempo que tarda un AGV en circular del punto A al punto B.
- $T_{BA}$ : Tiempo que tarda un AGV en circular del punto B al punto A.

Si añadimos múltiples tareas  $n$ , entra en juego la distribución de control de los AGV. En los siguientes apartados, se expondrá la metodología de cálculo del número mínimo de AGV necesario para cada distribución de AGV.

### 3.2.1.1 Cálculo analítico de ruta cíclica



*Figura 3.2.2:* Esquema básico de ruta cíclica

A la hora de gestionar los vehículos para un número de tareas,  $N_s$  (Número de tareas). Uno de los modelos más utilizados en la industria es el llamado *ruta cíclica*; en él, cada AGV tiene asignada únicamente una tarea, siendo posible que varios AGV estén asignados a la misma tarea. En este caso, el  $N_{AGV}$  siempre será igual o superior al  $N_s$ , sin importar los valores  $T_{e_n}$  (Periodo entre peticiones de la tarea  $n$ . Medido en segundos). Este cálculo se puede realizar mediante la siguiente fórmula:

$$N_{AGV} = \sum_{n=1}^{N_s} \left\lceil \frac{T_{AB_n} + T_{BA_n}}{T_{e_n}} \right\rceil \quad (3.2.2)$$

Donde:

- $N_s$ : Número de tareas.
- $T_{e_n}$ : Periodo entre peticiones de la tarea  $n$ . Medido en segundos.
- $T_{AB_n}$ : Tiempo que tarda un AGV en circular del destino  $A_n$  al destino  $B_n$  de la tarea  $n$ .
- $T_{BA_n}$ : Tiempo que tarda un AGV en circular del punto  $B_n$  al punto  $A_n$  de la tarea  $n$ .

**3.2.1.2 Cálculo analítico de punto de salida único**

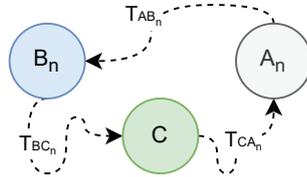


Figura 3.2.3: Esquema de ruta de un AGV en configuración con punto de salida único

Dependiendo de la morfología de la instalación, podemos usar un sistema de distribución de los AGV basado en punto de salida único, muy similar a una parada de taxi. En este caso, los AGV esperan en cola en  $C$ , y al AGV más adecuado se le asigna la misión de la tarea  $n$ , mediante la secuencia de destinos  $A_n$ ,  $B_n$  y  $C$ , indicado en la Figura 3.2.3. Por lo tanto, el número mínimo de  $N_{AGV}$  necesarios se calcula mediante la siguiente fórmula:

$$N_{AGV} = \left\lceil \sum_{n=1}^{Ns} \frac{T_{CA_n} + T_{AB_n} + T_{B_nC}}{Te_n} \right\rceil \tag{3.2.3}$$

Donde:

- $T_{CA_n}$ : Tiempo que tarda un AGV en circular del punto  $C$  al punto  $A_n$  de la tarea  $n$ .
- $T_{B_nC}$ : Tiempo que tarda un AGV en circular del punto  $B_n$  de la tarea  $n$  al punto  $C$ .

**3.2.1.3 Cálculo analítico de punto de salida múltiple**

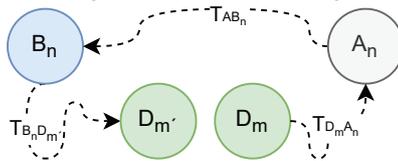


Figura 3.2.4: Esquema de ruta un AGV en configuración con punto de salida múltiple

En una instalación, se distribuyen diferentes puntos de descanso de los AGV, los cuales denominaremos  $D_m$ . Cuando un AGV recibe una misión y se encuentra en un punto de descanso  $D_m$ , se dirige al punto de inicio de la tarea, que será el punto  $A_n$ , y finaliza la tarea en el destino  $B_n$ . Posteriormente, el AGV se dirige

al punto de descanso  $D_{m'}$  que el Gestor de flotas considere más adecuado. Este comportamiento se puede observar en la Figura 3.2.4.

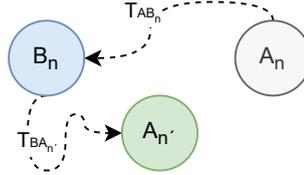
Por lo tanto, el cálculo mínimo de  $N_{AGV}$  es el siguiente:

$$N_{AGV} = \left[ \sum_{n=1}^{N_s} \frac{T_{D_m A_n} + T_{AB_n} + T_{B_n D_{m'}}}{Te_n} \right] \quad (3.2.4)$$

Donde:

- $T_{D_m A_n}$ : Tiempo que tarda un AGV en circular del  $D_m$  del punto de salida múltiple  $m$  al punto  $A_n$  de la tarea  $n$ .
- $T_{B_n D_{m'}}$ : Tiempo que tarda un AGV en circular del  $B_n$  de la tarea  $n$  al punto múltiple  $D_{m'}$  con ID  $m'$ .

#### 3.2.1.4 Cálculo analítico de distribución libre



**Figura 3.2.5:** Esquema de ruta un AGV en configuración con distribución libre

En este caso, utilizamos una distribución libre en la que los AGV, al llegar a su destino  $B_n$ , se dirigen directamente al inicio de la siguiente tarea  $A_{n'}$ . La secuencia de destinos queda así:  $A_n$ ,  $B_n$  y  $A_{n'}$ . Siendo  $A_{n'}$  el inicio de la tarea que aporte un mejor rendimiento. Por ello, el cálculo, contando con este factor, sería el siguiente:

$$N_{AGV} = \left[ \sum_{n=1}^{N_s} \frac{T_{AB_n} + T_{B_n A_{n'}}}{Te_n} \right] \quad (3.2.5)$$

Donde:

- $T_{B_n A_{n'}}$ : Tiempo que tarda un AGV en circular del destino  $B_n$  al destino  $A_{n'}$ , donde  $A_{n'}$  es el destino de recogida de la siguiente tarea, que es más óptima en cuanto al rendimiento de la Gestión de flotas.

El valor de  $T_{B_n A_{n'}}$  depende de la decisión acerca de cuál es la tarea más

adecuada para ser ejecutada. Es difícil determinar el valor mínimo, ya que depende de diferentes factores. Pero este puede acotar el límite inferior a 0 (si el inicio de la siguiente misión es el mismo que el final de la misión anterior). Teniendo esto en cuenta, el  $N_{AGV}$  mínimo para una instalación es el siguiente:

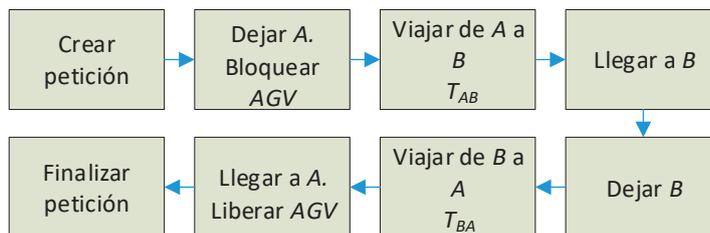
$$N_{AGV} = \left\lceil \sum_{n=1}^{N_s} \frac{T_{AB_n}}{T_{e_n}} \right\rceil \tag{3.2.6}$$

### 3.2.2 Metodologías de cálculo numérico mediante programas genéricos

Otra metodología para la extracción del número de AGV es mediante el cálculo numérico del entorno. La principal desventaja de la extracción utilizando las fórmulas del apartado anterior (*metodologías de cálculo analítico*) es que se simplifica el periodo de tiempo de entrega de las tareas  $T_{e_n}$  a un valor fijo. En la industria, este valor, aunque se suele mantener dentro de unos márgenes, es común que se vea alterado. A través de un cálculo numérico es más sencillo incorporar tiempos de  $T_{e_n}$  que varíen a lo largo del tiempo. A continuación, se expondrá una metodología para realizar el cálculo numérico del número de AGV necesarios para llevar a cabo ciertas tareas.

Se han realizado diferentes modelos con el objetivo de compararlos con las fórmulas analíticas de la sección anterior. Estos modelos se ejecutarán utilizando un software de simulación de eventos discretos, en concreto, el *software Arena*. En los siguientes apartados, se expondrán los modelos realizados con este programa para diferentes distribuciones de AGV. Este software de simulación se basa en la utilización de recursos para simular el entorno.

#### 3.2.2.1 Cálculo numérico de una tarea



**Figura 3.2.6:** Elementos que determinan el comportamiento del *software* de simulación de eventos discretos en el caso de una tarea

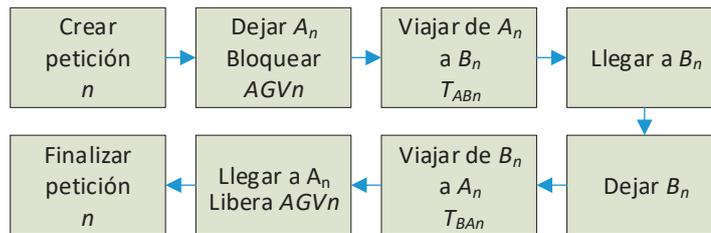
El modelo para una tarea sin restricción de tráfico se muestra en la Figura 3.2.6. Los bloques utilizados para emular el comportamiento de los AGV son los siguientes:

- **Crear petición.** Se genera en cada petición una nueva misión con una periodicidad de  $T_e$ , con la distribución requerida.
- **Dejar A.** Fase donde se utiliza un recurso AGV para realizar el transporte, si es que está disponible.
- **Viajar de A a B.** Con una duración de  $T_{AB}$ .
- **Llegar a B.** Se indica que se ha llegado al destino B.
- **Dejar B.** Se indica que se deja el destino B.
- **Viajar de B a A.** Con una duración de  $T_{BA}$ .
- **Llegar a A.** Se indica que se ha llegado al destino A liberando un recurso de AGV.
- **Finalizar petición.** Se hace indicando que se ha finalizado la misión y se utiliza para extraer las estadísticas de la tarea.

Para determinar el número de AGV necesarios, haremos el cálculo con un número elevado de recursos del tipo de los AGV. Este va decreciendo hasta obtener el número mínimo de AGV y seguir cumpliendo con las misiones en el tiempo establecido.

A continuación se mostrarán las metodologías para el cálculo de múltiples tareas utilizando el *software* descrito.

### 3.2.2.2 Cálculo numérico de la distribución cíclica



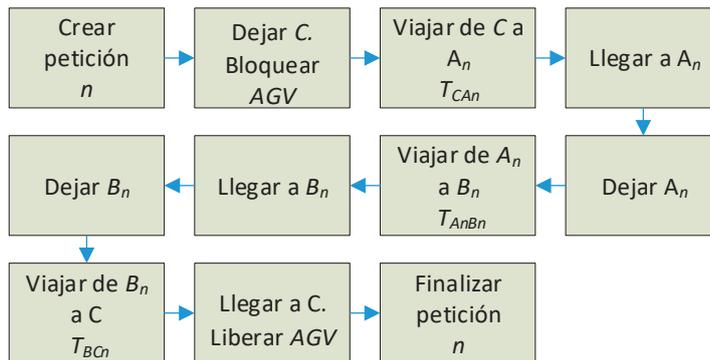
**Figura 3.2.7:** Bloques del *software* de simulación de eventos discretos para simular el caso cíclico sin restricciones

Para extrapolar el modelo de una única tarea para múltiples tareas y una distribución cíclica, realizaremos las modificaciones que se observan en la Figura

3.2.7. Seguimos la misma metodología, pero para cada una de las tareas, asignando un número determinado de recursos de AGV para cada tarea. En este caso, los bloques utilizados son los siguientes:

- **Crear petición de la tarea  $n$ .** Se genera en cada petición una nueva misión con una periodicidad de  $T_{e_n}$  según la tarea.
- **Dejar  $A_n$ .** Fase donde se utiliza un recurso AGV asignado a la tarea  $n$ , no pudiendo utilizar un recurso de AGV asignado a otra tarea.
- **Viajar de  $A_n$  a  $B_n$ .** Con una duración de  $T_{A_nB_n}$ .
- **Llegar a  $B_n$ .** Se indica que se ha llegado al destino  $B_n$ .
- **Dejar  $B_n$ .** Se indica que el AGV abandona el destino  $B_n$ .
- **Viajar de  $B_n$  a  $A_n$ .** Con una duración de  $T_{B_nA_n}$ .
- **Llegar a  $A_n$ .** Se indica que se ha llegado al destino  $A_n$  liberando un recurso de AGV asignado a la tarea  $n$ .
- **Finalizar petición.** Se hace indicando que se ha finalizado la misión y se utiliza para extraer las estadísticas de las tareas.

### 3.2.2.3 Cálculo numérico del caso de punto de salida único



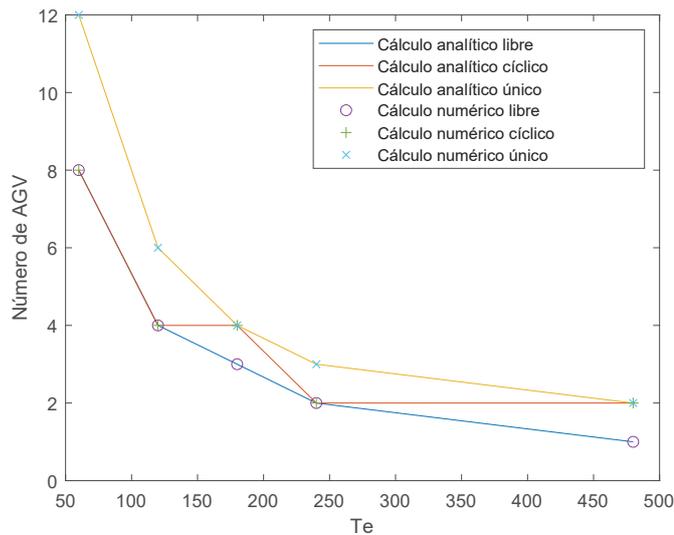
**Figura 3.2.8:** Bloques del *software* de simulación de eventos discretos para simular el caso de punto de salida único sin restricciones

Para el caso de punto de salida único, se modificarán los bloques añadiendo un destino extra. Pero en este caso, los recursos de los AGV son compartidos entre las tareas. Esto puede observarse en la Figura 3.2.8. Se lleva a cabo mediante las siguientes funciones:

- **Crear petición de la tarea  $n$ .** Se genera en cada petición una nueva misión

- con una periodicidad de  $Te_n$  según la tarea  $n$ .
- **Dejar C.** Fase en la que se utiliza un recurso AGV.
  - **Viajar de C a  $A_n$ .** Con una duración de  $T_{CA_n}$ .
  - **Llegar a  $A_n$ .** Se indica que se ha llegado al destino  $A_n$ .
  - **Dejar  $A_n$ .** Se indica que se abandona el destino  $A_n$ .
  - **Viajar de  $A_n$  a  $B_n$ .** Con una duración de  $T_{AB_n}$ .
  - **Llegar a  $B_n$ .** Se indica que se ha llegado al destino  $B_n$ .
  - **Dejar  $B_n$ .** Se indica que se abandona el destino  $B_n$ .
  - **Viajar de  $B_n$  a C.** Con una duración de  $T_{B_nC}$ .
  - **Llegar a C.** Se indica que se ha llegado al destino C liberando un recurso de AGV.
  - **Finalizar petición.** Indicando que se ha finalizado la misión, para poder extraer las estadísticas de las tareas.

### 3.2.2.4 Comparativa entre las metodologías de cálculo analítico y las metodologías de cálculo numérico



**Figura 3.2.9:** Comparativa del resultado del número de AGV necesarios entre las metodologías de cálculo analítico y las metodologías de cálculo numérico (Valor más bajo mejor)

Una vez expuestas las diferentes metodologías de cálculo numérico utilizando la herramienta de simulación por eventos discretos, procederemos a comparar el resultado con las metodologías de cálculo analítico para cada una de las diferentes distribuciones de los AGV, como se muestra en la Figura 3.2.9. Las distribuciones comparadas son las siguientes: distribución libre, distribución cíclica y

punto de salida único. Utilizaremos los siguientes parámetros:

- $T_{AB} = T_{BA} = T_{CA} = T_{BC} = 120$  segundos.
- Una única tarea.
- Calculamos los resultados con valores de  $Te$  con una distribución gaussiana. El valor medio de  $Te$  será de  $[480, 240, 180, 120, 60]$ .
- Simularemos 10 000 peticiones por cada rango de  $Te$  en la herramienta de simulación por eventos discretos.

Si analizamos los resultados en los tres tipos de escenarios de distribución de los AGV, observamos que el resultado de los cálculos es el mismo para las fórmulas extraídas mediante el *software* de simulación por eventos discretos y las fórmulas analíticas, por lo que el uso de este tipo de herramientas será necesario cuando el  $Te_n$  para cada tarea  $n$  no sea constante y varíe a lo largo del tiempo.

### 3.3 Algoritmo de cálculo dinámico del tiempo de trayecto

Las *metodologías de cálculo analítico* y *metodologías de cálculo numérico* son válidos para extraer el número mínimo necesario de AGV para las distribuciones de control de AGV expuestas. Pero estas metodologías tienen el inconveniente de que requieren conocer previamente el tiempo de trayecto necesario para circular de un destino a otro. Por este motivo, proponemos una metodología para calcular la ruta que tiene el menor tiempo de transporte y así calcular el tiempo que un AGV tarda en recorrer este trayecto.

Dicha metodología deberá ser capaz de calcular el tiempo mínimo de trayecto entre dos destinos. Este cálculo se realiza a partir del grafo del entorno y sus restricciones, contando con las características de los AGV. Nombrándola como la *metodología de cálculo del tiempo de trayecto mínimo*.

Este algoritmo se basa en el algoritmo *Dijkstra* [10], al que añadiremos el cálculo del desplazamiento entre los nodos, la aceleración y desaceleración del AGV, el tiempo extra para tomar cada curva y, en cada nodo, el tiempo de llegada según el nodo anterior.

Para calcular el tiempo que se tarda en ir de un punto a otro, procederemos a exponer el método de cálculo necesario. Para ello, contamos con la siguiente información: el nodo de inicio  $N_{Inicio}$ , el nodo de destino final  $N_{Final}$ , el grafo del entorno *GrafoMapa* y la información dinámica del AGV *Información<sub>AGV</sub>*. Estos son los pasos para poder hacer el cálculo:

1. Crear el grafo *GrafoMapaResol* a partir del *GrafoMapa*, el cual utilizaremos para calcular el tiempo del trayecto.
2. Resolver el *GrafoMapaResol* desde el  $N_{Inicio}$  al  $N_{Final}$  teniendo en cuenta la *Informacin<sub>AGV</sub>* y el *GrafoMapa*. Lo llevaremos a cabo mediante un algoritmo basado en *Dijkstra* [10].
3. Extraer del *GrafoMapaResol* el tiempo requerido para alcanzar  $N_{Final}$ .

A continuación se expone la metodología para resolver estos pasos. Posteriormente se expondrán las metodologías de cálculo del tiempo de trayecto de un segmento que une dos nodos, junto con un ejemplo ilustrativo de estas metodologías.

### 3.3.1 Generación del grafo



[Redacted text]

[Redacted text]

[Redacted text]

[Redacted text]

[Redacted text block]

**3.3.2 Resolución del grafo**

[Large redacted text block]

[Redacted text block]

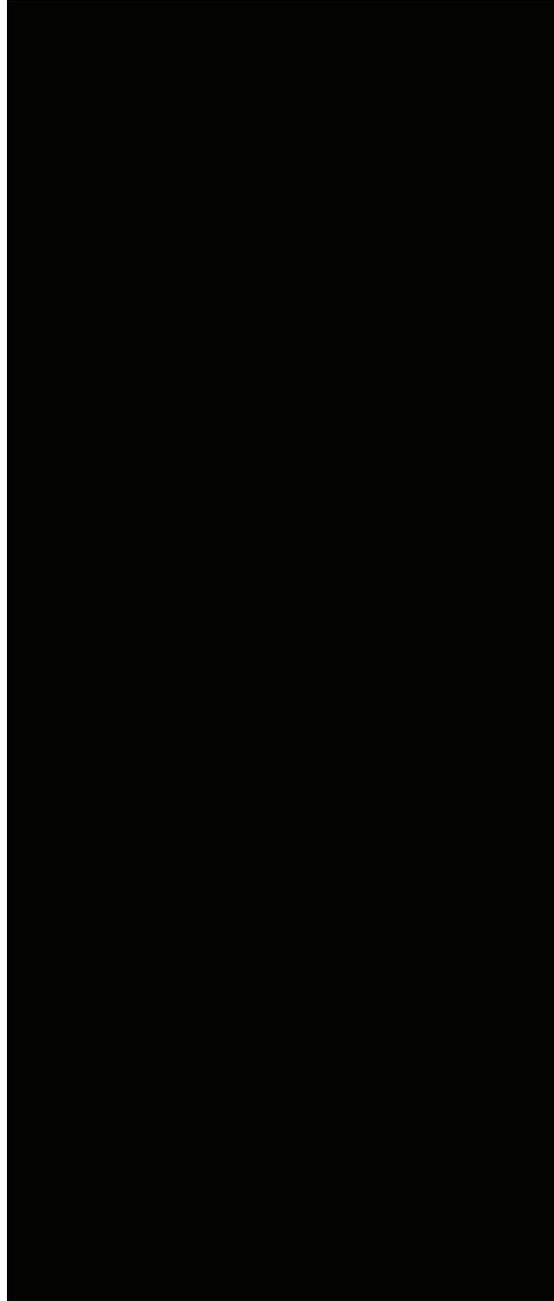
**3.3.3 Extracción del tiempo de navegación**

[Redacted text block]

**3.3.4 Cálculo del tiempo de trayecto de un segmento que une dos nodos**

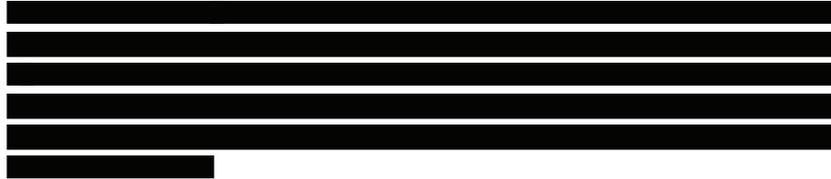
[Redacted text block]

### 3.3.5 Ejemplo de extracción del tiempo de navegación



*Figura 3.3.3:* Diagrama ejemplo de la resolución del grafo de la ruta

[Redacted text block]



### 3.4 Cálculos con restricción de tiempo de operación y carga de baterías

Casi la totalidad de los AGV se mueven mediante energía eléctrica suministrada por acumuladores energéticos, que requiere absorber energía eléctrica durante el proceso. Dependiendo del entorno, la estrategia de absorción energética puede variar. Las más utilizadas son las siguientes:

1. **Cambio de baterías.** Se cambian manualmente las baterías de los AGV. Era una práctica muy habitual hace unos años, cuando la tecnología de las baterías no permitía cargas de baterías con altos ratios de potencia. Actualmente, su uso es minoritario.
2. **Carga de baterías en movimiento.** Existe una zona en la que los AGV, al circular por ella, se cargan. Esta metodología, aunque implica un gran rendimiento, tiene el enorme inconveniente de tener un alto coste de implantación. Normalmente, se utiliza en los AGV en los que la velocidad máxima es muy baja debido a los requerimientos del proceso, y donde se agrupan muchos AGV en poca distancia.
3. **Carga de baterías puntual por oportunidad.** Los AGV cargan las baterías cuando se encuentran detenidos a causa de un proceso de la instalación, ya sea cuando llegan al punto de recogida o de entrega de material. Esto se aprovecha del tiempo de espera generado por el proceso de la instalación, por lo que se coloca un cargador de baterías puntual para poder cargarlos.
4. **Zonas de carga de baterías.** Los AGV realizan su trabajo y, dependiendo del nivel de carga de sus baterías, se dirigen a una zona exclusiva de carga de baterías hasta obtener la carga necesaria.

A continuación se muestran las metodologías necesarias para incluir las estrategias más comunes de carga de baterías **Carga de baterías puntual por oportunidad** y **Zonas de carga de baterías**.

Los casos más comunes de metodología de carga son la **Carga de baterías puntual por oportunidad** y **Zonas de carga de baterías**, procediendo a exponer, en los siguientes apartados, los diferentes cálculos derivados de añadir al sistema las restricciones de carga de baterías mediante estas dos metodologías.

#### 3.4.0.1 Carga de baterías por oportunidad

En la carga por oportunidad se aprovechan los tiempos de espera para cargar las baterías de los AGV. Normalmente, con esta estrategia, los AGV aprovechan las dos características que explicamos a continuación y con las que suelen contar las instalaciones:

- **Tiempo de operación.** En los destinos, debido al proceso que se da en la instalación, puede ser que un AGV se encuentre detenido, usualmente realizando la carga y descarga del material a transportar. La mejor estrategia para la colocación del punto de carga de baterías es situarlo en una posición en la que el AGV se encuentre detenido durante más tiempo.
- **Tiempo de espera de recepción de nueva misión.** Normalmente, un AGV realiza las misiones con un margen superior al requerido. Por ese motivo, estas posiciones de espera pueden ser una localización adecuada para la carga de baterías.

Utilizando estas características, podemos extraer el tiempo de espera  $TEsp$  de los AGV en la posición escogida. Se calcula mediante la siguiente fórmula:

$$TEsp = \frac{Te}{N_{AGV}} - TTM + TPro_{desp} - \sum_{d=1}^{Nd} TPro_d \quad (3.4.1)$$

Donde:

- $TEsp$ : Tiempo en que un AGV se encuentra esperando una nueva misión.
- $TTM$ : Tiempo de desplazamiento un AGV entre todos los destinos de la misión.
- $TPro_d$ : Tiempo que un AGV tiene que permanecer en el destino  $d$  de una tarea, ya sea por el tiempo requerido por el vehículo para realizar una tarea, o por los requerimientos de la instalación.
- $TPro_{desp}$ : Tiempo de operación que tarda un AGV en realizar una tarea en el lugar donde le espera una nueva misión.

Para añadir la restricción de la carga de baterías, en primer lugar, debemos calcular el tiempo necesario para cargar las baterías de un AGV y poder añadir a sus acumuladores la energía que ha consumido al realizar la misión anterior, asegurándonos de que la capacidad energética de los acumuladores sea superior

a la energía necesaria para realizar la siguiente misión. Para ello, es necesario contar con las características de los sistemas involucrados. El tiempo de carga de baterías se determina mediante la siguiente fórmula:

$$TBat = \frac{EDesp * \mu Bat}{PBat} + TIni \quad (3.4.2)$$

Donde:

- *TBat*: Tiempo requerido para la carga de la batería de un AGV.
- *EDesp*: Energía consumida por un AGV desde su última carga de baterías.
- *μBat*: Eficiencia de carga de la batería, contando con todos los elementos que intervienen en el proceso de absorción de energía por parte de un AGV.
- *TIni*: Tiempo de inicialización de la carga. Los sistemas de carga de baterías requieren un tiempo de inicialización para comenzar a cargar las baterías.
- *PBat*: Potencia de carga del cargador, suponiendo que la carga es lineal. (En muchas baterías la potencia de carga es constante entre los rangos de trabajo comunes [entre 20 % y 80 %]).

En esta estrategia de carga por oportunidad, una vez contemos con el *TBat* y el *TEsp*, podemos determinar que la carga de baterías afectará incrementando el número de AGV necesarios, siempre que se cumpla la función:

$$TBat > TEsp \quad (3.4.3)$$

Si tenemos en cuenta las restricciones de carga de baterías y de tiempo de operación, la expresión del cálculo del número de AGV necesarios mediante fórmulas analíticas para una tarea es la siguiente:

$$N_{AGV} = \left\lceil \frac{T_{AB} + T_{BA} + TBat - TPro_{desp} + \sum_{d=1}^{Nd} TPro_d}{Te} \right\rceil \quad (3.4.4)$$

Donde:

- *Nd*: Número de destinos de una tarea.

Aplicando esta restricción a las diferentes distribuciones de los AGV para *Ns* tareas, las fórmulas resultantes se muestran en los siguientes apartados.

### 3.4.0.1.1 Número de AGV con distribución cíclica

$$N_{AGV} = \sum_{n=1}^{Ns} \left[ \frac{T_{AB_n} + T_{BA_n} + T_{Bat_n} - T_{Prodesp_n} + \sum_{d=1}^{Nd_n} T_{Prod_n}}{Te_n} \right] \quad (3.4.5)$$

Donde:

- $Nd_n$ : Número de destinos de una tarea  $n$ .
- $T_{Prod_n}$ : Tiempo que un AGV tiene que permanecer en el destino  $d$  de la tarea  $n$ , ya sea por el tiempo requerido por el vehículo para hacer una tarea, o por los requerimientos de la instalación.

#### 3.4.0.1.2 Número de AGV con distribución de punto de salida único

$$N_{AGV} = \left[ \sum_{n=1}^{Ns} \frac{T_{CA_n} + T_{AB_n} + T_{B_nC} + T_{Bat_n} + \sum_{d=1}^{Nd_n} T_{Prod_n}}{Te_n} \right] \quad (3.4.6)$$

#### 3.4.0.1.3 Número de AGV con distribución de punto de salida múltiple

$$N_{AGV} = \left[ \sum_{n=1}^{Ns} \frac{T_{D_mA_n} + T_{AB_n} + T_{B_nD_{m'}} + T_{Bat_n} + \sum_{d=1}^{Nd_n} T_{Prod_n}}{Te_n} \right] \quad (3.4.7)$$

#### 3.4.0.1.4 Número de AGV con distribución libre

$$N_{AGV} = \left[ \sum_{n=1}^{Ns} \frac{T_{AB_n} + \max(T_{Bat_n}, \sum_{d=1}^{Nd_n} T_{Prod_n})}{Te_n} \right] \quad (3.4.8)$$

#### 3.4.0.2 Zonas de carga de baterías

En la estrategia de colocar zonas de carga de baterías, a diferencia de la estrategia de carga por oportunidad, la carga no se realiza en los tiempos de espera de los AGV, sino que cuando se les descargan las baterías, los AGV se dirigen a

estas zonas de carga de baterías y permanecen allí hasta que han recargado un porcentaje considerable de carga.

Por ello, los AGV no realiza la carga de baterías después de cada misión, sino que realiza varias misiones antes de dirigirse a una de estas zonas para recargar las baterías. El tiempo que un AGV pasa recargando sus baterías es el siguiente:

$$TBatZona = \frac{\left(\sum_{m=1}^{Nm} EDesp_m\right) * \mu Bat}{PBat} + TIni + TTBatZona \quad (3.4.9)$$

Donde:

- $TBatZona$ : Tiempo de carga de baterías requerido desde la última carga.
- $EDesp_m$ : Energía consumida por las misiones  $m$ .
- $TTBatZona$ : Tiempo de desplazamiento a la zona de carga de baterías.
- $Nm$ : Número de misiones realizadas desde la última carga de baterías.

La principal ventaja de esta distribución es que el porcentaje de uso de los cargadores es mayor, haciendo que el número de cargadores necesarios sea menor. Aunque, al precisar zonas separadas, requiere mayor espacio.

Cuando se aplica esta metodología de carga de baterías, la restricción de tiempo de operación y carga de baterías para cada distribución de AGV se muestra en los siguientes apartados.

#### 3.4.0.2.1 Número de AGV necesarios con distribución cíclica

$$N_{AGV} = \sum_{n=1}^{Ns} \left[ \frac{T_{AB_n} + T_{BA_n} + \frac{TTBatZona_n}{Nm_n} + \sum_{d=1}^{Nd_n} TProd_n}{Te_n} \right] \quad (3.4.10)$$

- $TTBatZona_n$ : Tiempo de desplazamiento a la zona de carga de baterías requerido al realizar la tarea  $n$ .
- $Nm_n$ : Número de misiones realizadas desde la última carga de baterías de una tarea  $n$ .

#### 3.4.0.2.2 Número de AGV con distribución de punto de salida único

$$N_{AGV} = \left[ \sum_{n=1}^{N_s} \frac{T_{CA_n} + T_{AB_n} + T_{BC_n} + \frac{TTBatZona_n}{N_{m_n}} + \sum_{d=1}^{N_{d_n}} TPro_{d_n}}{Te_n} \right] \quad (3.4.11)$$

### 3.4.0.2.3 Número de AGV necesarios con distribución de punto de salida múltiple

$$N_{AGV} = \left[ \sum_{n=1}^{N_s} \frac{T_{D_m A_n} + T_{AB_n} + T_{B_n D_{m'}} + \frac{TBatZona_n}{N_{m_n}} + \sum_{d=1}^{N_{d_n}} TPro_{d_n}}{Te_n} \right] \quad (3.4.12)$$

### 3.4.0.2.4 Número de AGV con distribución libre

$$N_{AGV} = \left[ \sum_{n=1}^{N_s} \frac{T_{AB_n} + \max\left(\frac{TTBatZona_n}{N_{m_n}}, \sum_{d=1}^{N_{d_n}} TPro_{d_n}\right)}{Te_n} \right] \quad (3.4.13)$$

## 3.5 Extracción del rendimiento máximo mediante cálculo

Con las metodologías expuestas, es posible desarrollar un método para calcular los AGV necesarios para realizar tareas en un entorno específico utilizando una distribución particular de AGV. Para ello, se deben seguir los siguientes pasos:

1. **Cálculo del tiempo de trayecto:** Se debe calcular el tiempo de trayecto para todas las tareas utilizando la metodología descrita en la Sección 3.3.
2. **Cálculo del número de AGV requerido:** Para esta fase, se pueden utilizar métodos analíticos o numéricos. Si el valor del tiempo de petición entre tareas es constante, se pueden utilizar los métodos analíticos descritos en la Sección 3.2.1 o las fórmulas de la Sección 3.4, si se quiere introducir restricciones de tiempo de operación o de cargar de baterías. Si el tiempo entre peticiones de las tareas no es constante, se debe utilizar el método numérico descrito en la Sección 3.2.2. Para aplicar cualquiera de

las metodologías, se debe tener en cuenta la correspondiente utilización para a la distribución de AGV utilizada.

Aunque esta metodología puede ser utilizado para calcular el rendimiento máximo, tiene la limitación de requerir una operación manual para su cálculo. por ese motivo, en el Capítulo 4, se presentan otras metodologías completamente automatizadas para el cálculo.

## Capítulo 4

# Cálculo de rendimiento de los Gestores de flotas mediante simulación

### Resumen

En este capítulo se presenta una metodología para calcular el rendimiento de los vehículos a través de un simulador desarrollado específicamente para esta tesis. El simulador es un emulador del comportamiento dinámico de los AGV, en el que los AGV virtuales siguen las directrices proporcionadas por diferentes algoritmos de Gestión de flotas programados en él.

El simulador permite evaluar el rendimiento de una flota de AGV en distintos escenarios y bajo diferentes condiciones de operación. De esta manera, es posible analizar el impacto de distintas estrategias de Gestión de flotas en la eficiencia y reducción de costes del sistema.

Además, el simulador ofrece la posibilidad de visualizar una simulación mediante un entorno en 3D dimensiones, lo que permite observar los puntos de conflicto del sistema.

En resumen, el simulador desarrollado para esta tesis es una herramienta

para el análisis y la evaluación del rendimiento de los vehículos en un sistema de transporte automatizado basado en AGV gobernados por un Gestor de flotas. Adicionalmente, también se permite extraer el rendimiento máximo teórico.

La correcta correlación entre el simulador y el mundo real se ha corroborado con datos reales del Entorno 2 propuesto en la Sección 6.3 y con el entorno de laboratorio real expuesto en la Sección 6.5.

Los apartados expuestos para exponer el simulador en este capítulo son los siguientes:

- **Características y restricciones del simulador:** En este apartado se exponen las características del simulador, incluyendo las restricciones y supuestos utilizados en la simulación, tales como las limitaciones por el tráfico, los tiempos de carga y operación, y otros factores relevantes.
- **Datos requeridos:** En este apartado se exponen los datos necesarios para realizar una simulación, incluyendo información sobre la instalación, los vehículos autónomos y las tareas a realizar, entre otros.
- **Datos de salida del simulador:** En este apartado se exponen los datos de salida proporcionados por el simulador, los cuales pueden ser utilizados para evaluar el rendimiento y los costes asociados con la simulación.
- **Metodología de cálculo de rendimiento máximo mediante simulador:** En este apartado se describe la metodología para realizar una simulación y obtener como resultado el máximo valor teórico de optimización de costes.

## 4.1 Características y restricciones del simulador

Este simulador emula el comportamiento dinámico de los AGV, los cuales siguen rutas marcadas por el Gestor de flotas. Las principales características y restricciones del simulador son las siguientes:

- **Rutas de AGV superpuestas a un *Roadmap*:** Los AGV se mueven espacialmente siguiendo una ruta que discurre por un mapa (*Roadmap*) representado mediante un grafo.
- **Características dinámicas de los AGV:** Se tienen en cuenta las características dinámicas de los AGV, tales como la aceleración, la velocidad máxima y la masa del vehículo, entre otros.
- **Características del *Roadmap*:** El mapa se genera mediante segmentos que conectan los diferentes nodos que forman el mapa. En cada nodo se requiere información específica como la velocidad máxima permitida, el radio de giro, entre otros.
- **Detección de colisiones:** El simulador es capaz de emular la detección de colisiones que tienen los vehículos. Si dos vehículos fueran a colisionar, éstos se detendrán, como si estuvieran en el mundo real, cumpliendo con las normativas de seguridad. El Gestor de flotas no podrá realizar ninguna acción sobre ellos al entrar en este modo, tal y como ocurre en el mundo real.
- **Gestión de tareas:** El simulador incorpora un gestor de tareas que genera las peticiones según las especificaciones dadas. También se incorporan restricciones de las tareas, como el tiempo de llegada máximo permitido, el tiempo objetivo, entre otros.
- **Interfaz de usuario:** El simulador está equipado con una interfaz web que permite configurar todas las opciones sin necesidad de programación. Además de reducir el tiempo necesario para configurar una simulación, esta interfaz también facilita la visualización de los resultados, lo que resulta de gran ayuda para el análisis del rendimiento.
- **Recopilación de datos:** Durante la simulación, se registra el estado de la misma para generar estadísticas que permiten evaluar la reducción de costos producida por la flota de AGV controlada por un Gestor de flotas.
- **Visualización 3D:** El simulador utiliza un motor gráfico para representar el entorno en 3 dimensiones y mostrar el movimiento de los vehículos mediante sus modelos 3D. Esta herramienta permite observar fácilmente el comportamiento de los vehículos, identificar lugares de conflicto o error.
- **Consumo de energía:** El simulador calcula el consumo y la recarga de energía de las baterías de los AGV.
- **Requisitos del Gestor de flotas:** El marco de actuación de los Gestores de flotas es el siguiente:
  - Es conocedor del estado reportado por los vehículos virtuales.
  - Es conocedor de las tareas y misiones en curso y futuras.
  - Puede asignarle rutas a los AGV.

- Puede asignar AGV a las misiones.
- Puede detener cualquier AGV en cualquier momento.
- No puede mover vehículos que estén detenidos por detección de los sensores perimetrales anti colisión.

## 4.2 Datos requeridos

Con el objetivo de generar una simulación de flotas de AGV mediante el software específico desarrollado expresamente para extraer el rendimiento de un Gestor de flotas en un entorno, se hace necesario introducir en el simulador todos los datos necesarios para emular el comportamiento de los vehículos en dicho entorno. A continuación, se detallan los datos requeridos para realizar una simulación:

- **Los modelos de elementos industriales:** Antes de comenzar la definición de la simulación, es necesario introducir las características de los diferentes modelos industriales que se van a utilizar, tales como los modelos de los AGV, los modelos de los cargadores de baterías y los tipos de elementos a transportar.
- **El espacio de las instalaciones:** La definición del espacio de las instalaciones incluye varios elementos, como el *Roadmap*, las características de los puntos de entrega y la localización de los puntos de recarga de baterías.
- **El trabajo:** Las tareas corresponden al trabajo que deben realizar los AGV, donde se especifican los diferentes destinos que deben alcanzar, la periodicidad de las peticiones, el trabajo a realizar en cada destino, las restricciones de tiempo de entrega, entre otros aspectos.
- **Definición de la flota:** Una vez definido el entorno y el trabajo a realizar, el simulador requiere especificar el número de AGV de cada modelo (previamente especificados en el apartado *Los modelos de elementos industriales*) que se utilizarán.
- **Características de la simulación:** Finalmente, se requiere determinar los datos específicos de la simulación, que incluyen el tiempo de simulación, el algoritmo a utilizar y la posición de inicio de cada AGV.

### 4.2.1 Los modelos de elementos industriales

Para realizar una simulación de un sistema intralogístico que funcione mediante AGV, es fundamental contar con la definición de los elementos de hardware que se pretenden emular. Estos elementos están diseñados con características especificadas por su fabricante y su comportamiento debe ser emulado con precisión en el simulador. En este apartado se indican los datos requeridos para cada elemento.

Este simulador tendrá en cuenta los modelos de AGV, cargadores de baterías y los elementos a transportar. El elemento más importante es el AGV, re-

quiriendo información detallada.

Los datos utilizados para emular el comportamiento de los AGV son los siguientes:

- Dinámica del AGV:
  - Velocidad máxima del AGV.
  - Aceleración angular máxima del AGV.
  - Aceleración y desaceleración del AGV.
- Dimensiones del AGV.
- Dimensiones y posición de detección de los campos de seguridad de los diferentes sensores de seguridad del AGV (comportamiento para el cumplimiento de la normativa *ISO 3691-4* [19]).
- Tiempo necesario para arrancar el AGV (comportamiento para el cumplimiento de la normativa *ISO 3691-4* [19]).
- Características que afectan al consumo:
  - Masa del AGV.
  - Eficiencia del sistema motriz.
  - Consumo de los sistemas no motrices.
  - Fricción del vehículo al rodar.
  - Características de la batería: capacidad, voltaje, eficiencia de carga, etc.

El cargador de batería que se utilice afecta al tiempo de recarga de las baterías. En la Figura 4.2.1 se puede observar un ejemplo de especificación de un cargador. La información requerida para su correcta emulación en el simulador es la siguiente:

- Voltaje nominal.
- Voltaje máximo.
- Voltaje mínimo.
- Potencia de carga nominal.

**Editar cargador**

Nombre  
25A\_ChargerA

12 / 20

**Configuración**

Voltaje nominal (v)  
24

Voltaje max (v)  
28

Voltaje min (v)  
24

Energía de carga nominal (w)  
600

Figura 4.2.1: Ejemplo de definición de un cargador de baterías de un AGV

En cada instalación existen unos elementos a transportar. En la Figura 4.2.2 se puede observar un ejemplo de definición. Es imprescindible incluir estos elementos en la simulación, ya que afectan principalmente a los siguientes aspectos:

- El consumo de AGV se ve afectado por el peso del elemento a transportar.
- La detección de los elementos de seguridad se ve afectada si los elementos a transportar son más grandes que los AGV, lo que puede cambiar la detección anticolidión de los vehículos.

**Editar carga**

Nombre  
Load1000

8 / 20

**Configuration**

Tipo de carga  
OnBoard

Peso (kg)  
1000

Tamaño							
Posición de centro:	X (m)	Y (m)	Z (m)	Distancia frontal (m)	Distancia trasera (m)	Distancia derecha (m)	Distancia izquierda (m)
	0	0	0	1	1	1	1

Figura 4.2.2: Ejemplo de definición de un elemento a transportar

## 4.2.2 El espacio de las instalaciones

Otro elemento importante a definir es el espacio de las instalaciones donde se moverán los AGV. En este caso de uso, los vehículos se mueven a través de

grafos, por lo que para definir el *Roadmap* se utilizará una herramienta visual para definir los nodos y segmentos que lo forman. Los datos requeridos por cada nodo y cada segmento son los siguientes:

- **Por cada nodo:**
  - **Posición X y Y:** Según las coordenadas del mapa.
  - **Radio de giro:** Determina el radio de la ruta que realizarán los AGV entre un segmento de entrada y otro de salida.
  - **Conexiones de entrada al nodo,** determinadas por:
    - \* ID del segmento por el que entran los AGV.
    - \* ID del nodo del que procede el segmento.
  - **Conexiones de salida al nodo,** determinadas por:
    - \* ID del segmento por el que saldrán los AGV.
    - \* ID del nodo de destino del segmento.
  - **Velocidad máxima de circulación por el nodo.**
- **Por cada segmento:**
  - **ID del nodo de inicio del segmento.**
  - **ID del nodo de finalización del segmento.**

Un ejemplo para definir un *Roadmap* se puede observar en las Figuras 4.2.3 y 4.2.4.

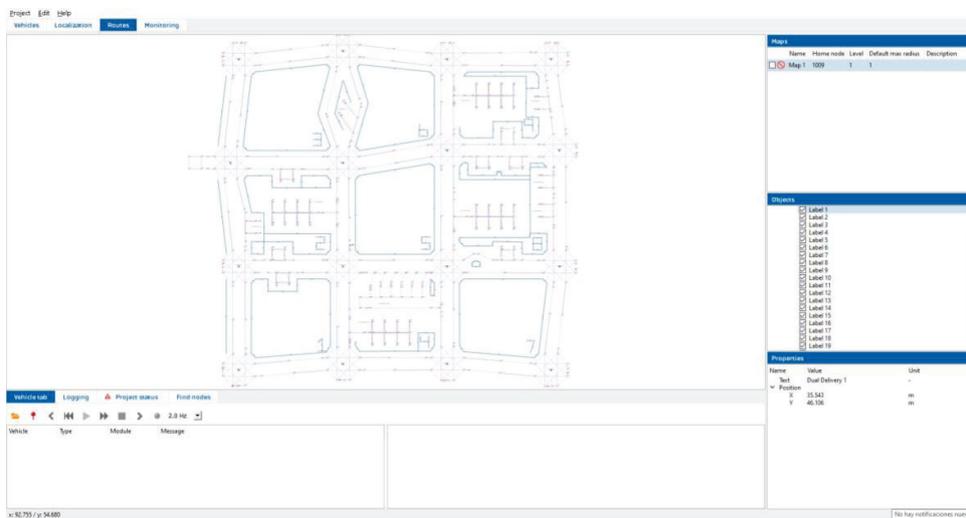


Figura 4.2.3: Ejemplo de definición de un *Roadmap*

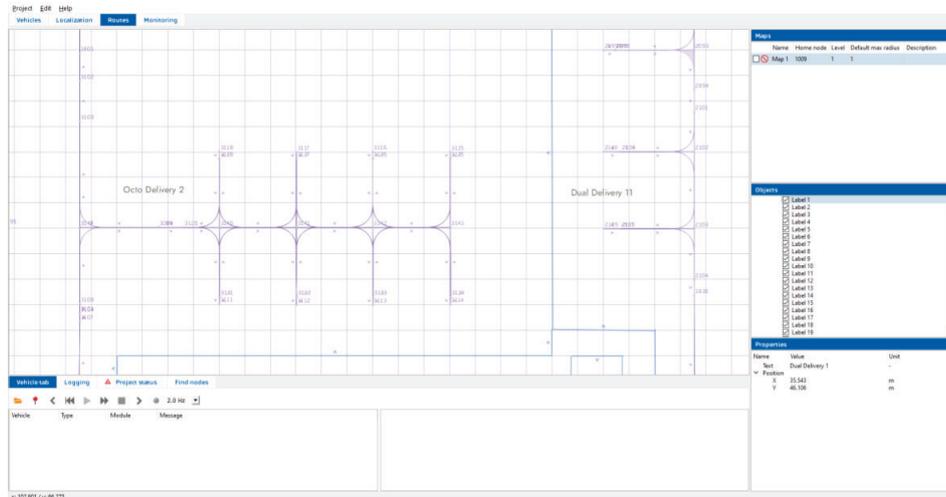


Figura 4.2.4: Detalle de ejemplo de definición de un Roadmap

Es común en la intralogística con vehículos autónomos que los puntos de entrega y recogida tengan varias posiciones, donde el vehículo tiene que colocarse en una de ellas siguiendo una lógica dependiendo de la ocupación u otros factores. Por este motivo, para lidiar con esta casuística, se introduce el concepto de entrega múltiple. Se requiere que los vehículos ocupen las posiciones dentro de este destino con una lógica determinada. Los factores que determinan un punto de entrega múltiple se muestran en la Figura 4.2.5 y son los siguientes:

- **Id del punto de entrega.**
- **Tipo de distribución de los puntos de parada:** Esta puede ser paralela o en serie.
- **Lógica de selección de un vehículo para iniciar una misión:** Esta depende del Gestor de flotas a utilizar, ya que algunos no incorporan esta lógica y se le aplica una estándar. Las opciones más comunes son las siguientes:
  - **Aleatoria.**
  - **FIFO.**
  - **Porcentaje de la carga de batería más alta.**
  - **Porcentaje de la carga de batería más baja.**
- **Posición de entrada:** Se indica la lógica en la cual se tiene que dirigir el vehículo cuando entra al punto de entrega múltiple. Las opciones son las siguientes:
  - **Aleatoria:** Se dirigirá aleatoriamente a un punto de entrega que no esté ocupado.
  - **En orden ascendente:** Se dirigirá por orden de definición al punto de

entrega que no esté ocupado.

- **Condiciones de entrada:** Pueden existir condiciones de entrada al punto de acceso múltiple, como el estado de la batería, el modelo del AGV, y la carga que lleva el AGV.
- **Punto de entrada:** Posición por donde se entra a la zona de entrega múltiple, en esta posición se determina el destino final del AGV.
- **Punto de salida:** En el destino por el cual se determina que el AGV sale de la zona de entrega múltiple.
- **Destinos finales de entrega:** Son los múltiples destinos finales de los AGV.

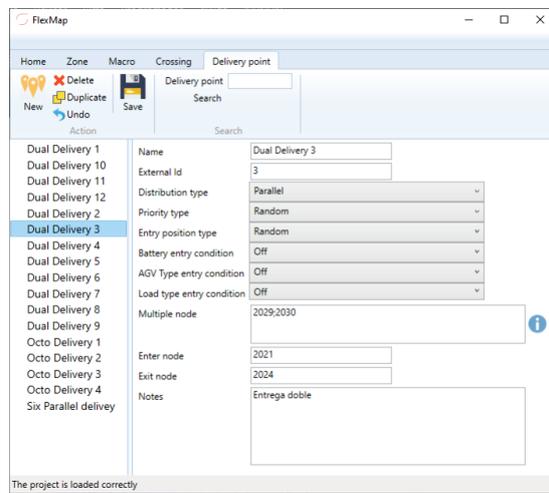


Figura 4.2.5: Detalle del un ejemplo de definición de puntos de entrega multiple

Los AGV, al funcionar mediante energía eléctrica proporcionada por baterías, requieren recargarse, siendo un factor determinante a tener en cuenta por el Gestor de flotas. Para abordar la carga de baterías de los AGV, se colocarán los cargadores de baterías en nodos dentro de un punto de entrega múltiple, donde el AGV puede tener que realizar otras tareas o no. El Gestor de flotas será conocedor de en qué puntos de entrega múltiple existen cargadores de baterías. Los modelos de cargadores de baterías especificados en el apartado 4.2.1 se utilizarán para este propósito. En la Figura 4.2.6 se muestra un ejemplo de la definición de un punto de recarga de baterías.

The screenshot shows a web interface for defining battery charging points. It features a 'TAREA' tab and a 'CARGADOR' section. In the 'CARGADOR' section, there is a 'NUEVO' button and a text input field containing 'Cargador nodo 3002'. Below this, there is a 'Nombre' field with the value 'Cargador nodo 3002'. A table below shows a list of charging points with columns 'Id', 'Cargador', and 'Nodo'. The table contains one row with the following data: Id: 1, Cargador: 25A\_ChargerA, and Nodo: 3002. At the bottom of the interface, there is an 'AÑADIR' button.

Id	Cargador	Nodo
1	25A_ChargerA	3002

**Figura 4.2.6:** Información requerida para definir los puntos de carga de batería que cuenta una instalación

### 4.2.3 El trabajo

Una vez determinados los entornos, se puede proceder a la definición de las tareas a realizar. En este documento, las tareas se exponen de forma determinada, tal como se indica en el apartado 2.2.3. Para definir el trabajo a realizar, se deben definir todas las tareas, como se muestra en el ejemplo de la figura 4.2.7. Posteriormente, como se ilustra en la figura 4.2.8, se indican los diferentes destinos de una tarea, junto con su tiempo de inicio, tiempo de fin y el intervalo de petición con la probabilidad correspondiente. Cabe destacar que los destinos de una tarea pueden corresponder a puntos de entrega múltiple, lo que implica la necesidad de seguir una lógica determinada para su ejecución, tal como se describe en el apartado anterior.

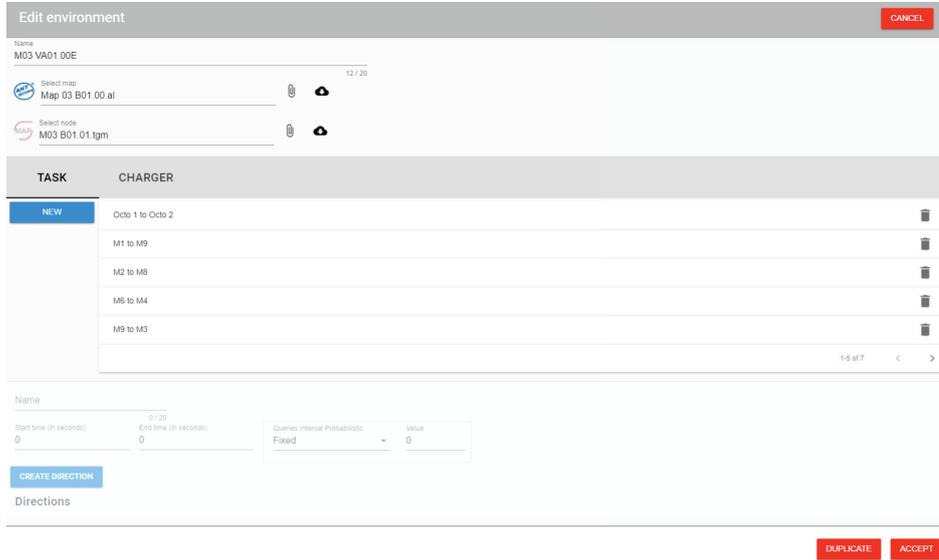


Figura 4.2.7: Ejemplo de definición de unas tareas dentro de un entorno



Figura 4.2.8: Ejemplo de definición de una tarea

En cada punto de entrega se pueden determinar las siguientes características:

- **Probabilidad de que este destino sea escogido:** Algunas instalaciones, puede ser que un porcentaje de veces tenga que ir a un destino y otro a otro dentro de una misma tarea.
- **Tiempo máximo de llegada:** Este tiempo se cuenta desde que se realizó la petición de la tarea. Este es el tiempo máximo aceptado por la instalación para que llegue el vehículo.
- **Tiempo objetivo de llegada:** Este tiempo se cuenta desde que se realizó la petición de la tarea. La instalación espera que el vehículo llegue antes de este tiempo. Si el AGV llega después de este tiempo, es posible que conlleve penalizaciones por parte de la instalación, pero que no son graves.
- **Elementos de transporte:** Se indica si se va a recoger o dejar un elemento de transporte. Indicando el modelo de este.
- **Espera por carga de baterías:** Los vehículos si se quedan sin energías pueden causar colapso en una instalación, motivo por el cual puede ser requerido que un vehículo no pueda dejar un destino si el porcentaje de batería es inferior al mínimo requerido.
- **Tiempo de espera:** Se indica el tiempo de espera en un punto de entrega, este puede causado por la carga y descarga del material a transportar, por procesos de la instalación u otros factores.
- **ID de destino:** Se indica si es un punto de entrega múltiple o uno simple y su correspondiente ID.

En la Figura 4.2.9 se observa un ejemplo de definición de un destino.

Configuración

Probabilidad de desintaci...   Multiple Node

Tiempo máximo (s)   Parar en destinación

Tiempo objetivo (s)   Condición multinode job

Carga

Espera de carga de batería  Valor (% decimal)

Waiting probabilistic time  Value

0 - Multiple Id 2000 - Probabilidad 100%

Figura 4.2.9: Ejemplo de definición de un destino de una tarea

#### 4.2.4 Definición de la flota

EditFleet

Name  6 / 30

AGV Type  Quantity

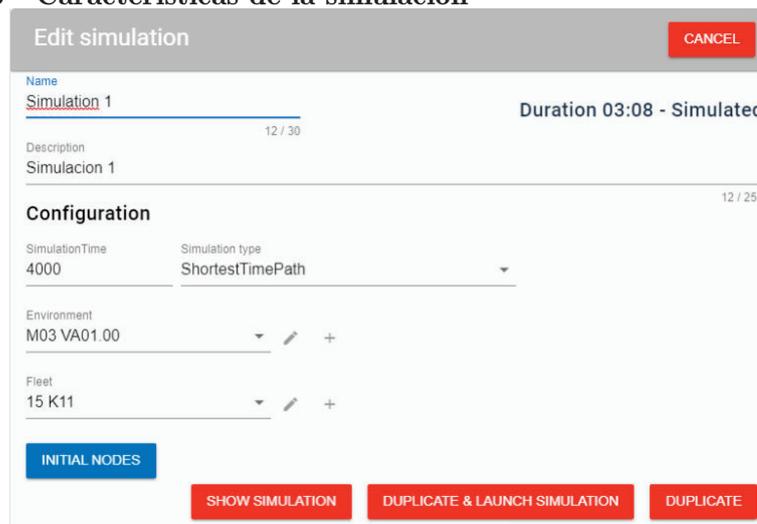
AGVs

<input type="checkbox"/>	Id	Type
<input type="checkbox"/>	15	K11 S420
<input type="checkbox"/>	14	K11 S420
<input type="checkbox"/>	13	K11 S420
<input type="checkbox"/>	12	K11 S420
<input type="checkbox"/>	11	K11 S420

Figura 4.2.10: Ejemplo de definición de una flota de AGV

La flota de AGV de la simulación se determina indicando el número de AGV de cada modelo que forman una flota. Como se puede observar en el ejemplo de la Figura 4.2.10, cada uno de los vehículos tiene asignada una ID para su identificación.

#### 4.2.5 Características de la simulación



*Figura* 4.2.11: Ejemplo de configuración de una simulación

Para realizar una simulación es necesario seleccionar un entorno con sus correspondientes tareas y asignar la flota de AGVs a emular. Para cada uno de los vehículos de la flota es necesario especificar su posición inicial, tal y como se muestra en el ejemplo de la Figura 4.2.12. Además, se deben proporcionar otros datos necesarios, como el tiempo de simulación requerido y la elección del algoritmo a utilizar. En la Figura 4.2.11 se puede observar un ejemplo de configuración de una simulación.

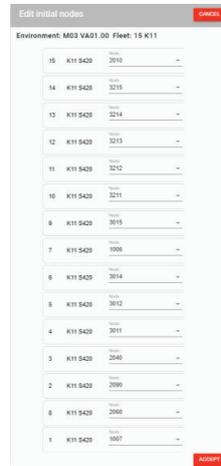


Figura 4.2.12: Ejemplo de la asignación de posición inicial de una flota de AGV

### 4.3 Datos de salida del simulador

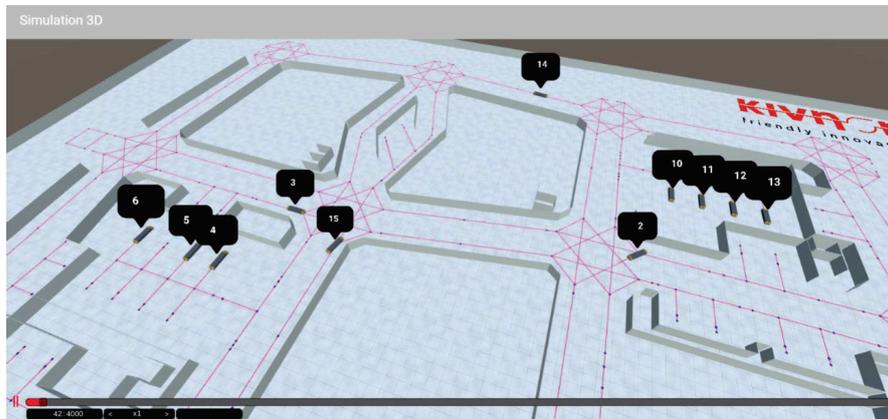


Figura 4.3.1: Ejemplo de la visualización 3D resultante de una simulación

Al realizar una simulación se obtienen los siguientes datos como resultados:

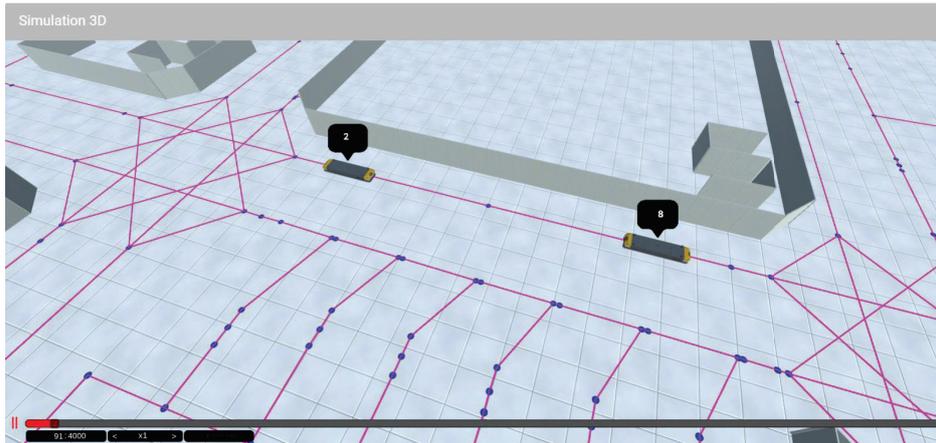
- **Visualización en 3D:** Se obtiene una visualización en 3D de la instalación, que permite navegar espacial y temporalmente por la misma, observando el movimiento de los AGVs y permitiendo identificar fácilmente los puntos con mayor congestión de tráfico. En las Figuras 4.3.1 y 4.3.2 se muestra

un ejemplo de esta visualización.

- **Estadísticas resultantes:** Estos son los siguientes:
  - Datos de la flota de AGV:
    - \* Tiempo total que han estado parados todos los AGV durante las misiones.
    - \* Tiempo total que han estado parados todos los AGV y no estaban en una misión.
    - \* Tiempo total que han estado acelerando todos los AGV.
    - \* Tiempo total que han estado desacelerando todos los AGV.
    - \* Tiempo total que han estado a velocidad constante todos los AGV.
    - \* Energía total consumida por todos los AGV.
    - \* Número de vehículos que han estado alguna vez en una misión.
  - Datos de cada AGV:
    - \* Tiempo total que ha estado parado el AGV durante las misiones.
    - \* Tiempo total que ha estado parados todos el AGV y no estaban en una misión.
    - \* Tiempo total que ha estado acelerando el AGV.
    - \* Tiempo total que ha estado desacelerando el AGV.
    - \* Tiempo total que ha estado a velocidad constante el AGV.
    - \* Energía total consumida por el AGV.
    - \* Estadísticas del estado de carga de la batería.
  - Datos de cada intersección:
    - \* Ratio de ocupación de AGV.
    - \* Ratio de paso de AGV.
  - Datos de cada destino múltiple:
    - \* Ratio de ocupación de AGV.
    - \* Ratio de paso de AGV.
  - Datos de cada cargador de baterías:
    - \* Ratio de ocupación de AGV.
    - \* Energía total suministrada.
  - Datos de cada tarea:
    - \* Número de peticiones realizadas.
    - \* Número de peticiones completadas.
    - \* Número de peticiones en proceso.
    - \* Utilización de AGV por la tarea (último, máxima, media y mínima).
    - \* Tiempos que se han tardado en completar las diferentes peticiones (último, máxima, media y mínima).
    - \* Diferencia de tiempo de llegada respecto al tiempo objetivo de cada destino (último, máxima, media y mínima).
    - \* Diferencia de tiempo de llegada respecto al tiempo máximo admisible de cada destino (último, máxima, media y mínima).
    - \* Número de veces que la tarea ha sido completada dentro del tiempo objetivo.
    - \* Número de veces que la tarea ha sido completada fuera del tiempo

objetivo.

- \* Número de veces que la tarea ha sido completada dentro del tiempo máximo admisible.
- \* Número de veces que la tarea ha sido completada fuera del tiempo máximo admisible.



*Figura 4.3.2:* Ejemplo de la visualización 3D resultante de una simulación, con la cámara cerca de los AGV

#### 4.4 Metodología de cálculo de rendimiento máximo mediante simulador

Para extraer el rendimiento máximo teórico, en vez de utilizar la metodología expuesta en la sección 3.5, la cual requiere un cálculo manual, se utilizará el simulador expuesto en este capítulo con las siguientes variaciones:

1. **Desactivación de las colisiones:** Dado que es el factor que más influye al rendimiento de una flota de AGV es el tráfico, se deshabilita la detección de colisión de vehículos y su posterior detención, para que no se tenga en cuenta el tráfico. De esta forma, se reduce la afectación del tráfico a 0 para este cálculo.
2. **Cálculo de rutas:** El algoritmo utilizado para esta metodología será el expuesto en la Sección 3.3.4, correspondiente a la ruta óptima sin considerar el tráfico.
3. **Desactivación de la gestión de intersecciones:** En una intersección, la mejor gestión teórica posible es que todos los vehículos puedan cruzar sin ningún retraso, lo cual podría causar colisiones. No obstante, al estar deshabilitada la detección de colisiones, se puede extraer así el mejor

rendimiento teórico posible mediante la simulación.

De esta forma, se puede extraer el rendimiento máximo posible. En este documento, el rendimiento máximo teórico será nombrado como *STPI*.

Se ha validado el correcto funcionamiento del simulador comparando los resultados obtenidos mediante la simulación con los datos reales del Entorno 2 propuesto en la Sección 6.3 y con el entorno de laboratorio real expuesto en la Sección 6.5. Se realizaron comparaciones en 5 escenarios distintos y se observó que la desviación en todos los casos fue inferior al 5%.

## Capítulo 5

# Estado del arte

### Resumen

Este capítulo presenta diferentes algoritmos de Gestión de flotas con el objetivo de evaluarlos y obtener valores de referencia de rendimiento. Estos algoritmos se denominarán como algoritmos de referencia en esta tesis. Como primer paso, se proporcionará una breve clasificación de las diferentes categorías de algoritmos para seleccionar los más significativos de cada tipología. Las categorías se han clasificado de la siguiente forma: los convencionales, que son algoritmos comúnmente utilizados en la industria debido a su fácil integración; los no convencionales, que son algoritmos basados en referencias científicas, pero que no son comunes en la industria actualmente; y, finalmente, se seleccionará un algoritmo ideal para conocer el límite máximo de reducción de costos.

Los apartados que forman este capítulo son los siguientes:

- **Introducción:** En este apartado se exponen diferentes métodos de clasificación de algoritmos de Gestión de flotas, seleccionando los algoritmos más significativos de cada tipología.
- **Algoritmo de límite de rendimiento:** Se expone el algoritmo utilizado para determinar el límite ideal máximo de rendimiento.
- **Algoritmos convencionales:** Se exponen los algoritmos más comúnmente utilizados en la industria.
- **Algoritmos no convencionales:** Se exponen algoritmos expuestos en artículos científicos, pero que aún no se conoce su aplicación en la industria, y

que podrían conllevar un mayor rendimiento que los algoritmos convencionales.

## 5.1 Introducción

Existen muchos tipos de algoritmos de Gestión de flotas, desde los más simples, que se pueden llevar a cabo manualmente usando una ruta fija, hasta otros algoritmos que son capaces de manejar rutas en tiempo real. Una manera de clasificarlos puede ser hacerlo según la distribución del control del Gestor de flotas:

- **Algoritmos descentralizados.** Los algoritmos descentralizados: [30], [1], [44] y [45] se caracterizan por tener un control por zonas o tareas, lo cual facilita la escalabilidad y la implantación. Estos son los más comúnmente utilizados en la industria. Este control por zonas o tareas genera que los algoritmos solo tengan una visión local lo cual implica un menor rendimiento en entornos complejos. Existen otros algoritmos: [11] y [29], que intentan resolver este problema, pero como solo tienen en cuenta aspectos locales, no ofrecen los mejores resultados finales.
- **Algoritmos centralizados.** Por otro lado, los algoritmos que disponen de toda la información del sistema: [25], [8] y [17] son más complejos, ya que deben conocer el estado total del sistema para, así, poder obtener un mayor rendimiento. Pero eso provoca que su escalabilidad sea más costosa, por lo que su instalación conlleva un mayor coste.
- **Algoritmos jerárquicos.** La utilización de las estrategias jerárquicas: [37], [9] y [7] intenta explotar los beneficios de los sistemas centralizados y descentralizados, minimizando las desventajas de cada uno de ellos. Estas estrategias combinan algoritmos centralizados y descentralizados.

Una vez expuestas las diferentes tipologías de distribución de control, otro factor que diferencia el rendimiento de los algoritmos de una Gestión de flotas es la metodología del cálculo de la ruta más óptima. Las más extendidas son las que se basan en los siguientes algoritmos:

- **Rutas Fijas.** Se trata de una metodología muy extendida que muchas veces viene predefinida por el entorno. La ventaja de esta metodología, aparte de su fácil implementación, es que se puede segregarse el tráfico al seleccionar la ruta de cada destino, ya que en muchas ocasiones, la ruta óptima individual no es la ruta óptima si la consideramos de forma global.
- **Dijkstra.** Algunos de los métodos más comúnmente utilizados para calcular la ruta óptima en un *Roadmap* están basados en *Dijkstra* [2]. La ventaja es que, con esta base, se le pueden añadir otras consideraciones [40]. Uno de los objetivos será tener en cuenta el tráfico a la hora de calcular la ruta, y una de las metodologías puede ser la incorporación de ventanas de tiempo [13], [15], [5], [6], y [4].
- **A\* (A Star).** Otra vertiente de los algoritmos de búsqueda de rutas: [20]

y [26] son los que están basados en  $A^*$ . La ventaja del uso de este tipo de algoritmos es que su tiempo computacional es inferior a los basados en *Dijkstra*, sobre todo en mapas grandes. Estos pueden ser utilizados sin la necesidad de contar con un *Roadmap*.

Otros algoritmos: [13] y [39], intentan integrar una estratégica dinámica de coordinación. Por otro lado, muchos de los algoritmos implementan el uso de los *Roadmap*: [21] o [18] para simplificar el modelizado del entorno, así como para simplificar, en gran medida, los algoritmos que se van a implementar.

Existen diversos algoritmos de Gestión de flotas que pueden utilizarse como referencia, con varias formas de clasificarlos. En el sector industrial, la robustez es de gran importancia, siendo también un factor relevante en la Gestión de flotas. Esto conduce a que los algoritmos más comúnmente utilizados sean los más deterministas, proporcionando así un comportamiento y, por ende, un rendimiento más uniforme. La determinación de rutas es el factor que confiere a un algoritmo su carácter determinista, es decir, que para circular del destino  $A$  al  $B$ , la ruta es siempre la misma. Por lo tanto, clasificaremos este tipo de algoritmos como convencionales. Sin embargo, existen otros algoritmos menos deterministas, que dependiendo de diversos factores, usualmente el tráfico, pueden generar diferentes rutas para circular del destino  $A$  al  $B$ . A pesar de tener el potencial para ofrecer un rendimiento superior, estos algoritmos presentan una mayor incertidumbre. Así, clasificaremos estos Gestores de flotas como no convencionales. Actualmente, su uso en la industria no es extenso, pero están ganando terreno debido a su mayor potencial de rendimiento.

A continuación se muestran los algoritmos de referencia seleccionados, clasificándolos por su naturaleza en los siguientes grupos:

- **Algoritmo del límite de rendimiento:** En este apartado, planteamos una metodología orientada a determinar en cada escenario el límite superior teórico del rendimiento máximo. Este valor será utilizado como referencia de rendimiento máximo. El algoritmo utilizado para obtener este límite es el siguiente:
  - *Shortest Time Path with Ideal Traffic Considerations (STPI)*.
- **Algoritmos convencionales:** Se expondrán diversas metodologías que se corresponden a los algoritmos convencionales comúnmente implementados en la industria. Estos se fundamentan en rutas preestablecidas y la aplicación de lógicas puntuales. Los algoritmos seleccionados para esta tesis son los siguientes:
  - *Shortest Time Path with Random Priority Crossing (STPRAN)*.

- *Shortest Time Path with FIFO Priority Crossing (STPFIFO)*.
- *Shortest Time Path with Accumulative Priority Crossing (STPACC)*.
- *Shortest Time Path with Collision Table Crossing (STPCT)*.
- **Algoritmos no convencionales:** Se emplearán distintos algoritmos que ilustran el rendimiento de varias metodologías de gestión de flotas complejas. Estos no se implementan comúnmente en la industria, ya que todavía se encuentran en fase de investigación, según se indica en diversos artículos académicos, o se aplican en contextos aislados debido a su inherente complejidad. Se caracterizan por incorporar consideraciones del tráfico en la formulación de sus cálculos de ruta. Los algoritmos examinados son los siguientes:
  - *Dynamic Geodesic Environment Partition (GEOPAR)*.
  - *Conflict-free Multi-AGV Routing based on Improved A\* Algorithm (ASTAR)*.
  - *Dynamic Routing of Automated Guided Vehicles in Real-time (TIMEWINDOW)*.

## 5.2 Algoritmo de límite de rendimiento

Con este algoritmo, pretendemos extraer el límite máximo de rendimiento. La propuesta omitirá todas las consideraciones de tráfico y las posibles detenciones por las paradas de los vehículos para evitar colisiones. Por otro lado, será necesaria la utilización de las rutas que desemboquen en un mayor rendimiento sin tener en cuenta el factor del tráfico. Para ello, utilizaremos la metodología expuesta en el apartado del cálculo del rendimiento máximo del Capítulo 4. Concretamente, la expuesta en el epígrafe 4.4. Con esta metodología, podemos extraer la ruta más corta sin tráfico. A este algoritmo lo denominaremos *Shortest Time Path with Ideal Traffic Considerations (STPI)*.

## 5.3 Algoritmos convencionales

En el contexto de la industria de la Gestión de flotas, se utilizan diferentes tipos de algoritmos para planificar y gestionar las rutas de los AGVs. Estos algoritmos se caracterizan por definir rutas preestablecidas, que pueden ser calculadas de forma manual o automática, sin tener en cuenta las condiciones de tráfico [2]. Respecto a la implementación de estos algoritmos para el presente estudio, se hace uso de la metodología detallada en la Sección 3.3 con el propósito de calcular las rutas. Dicha metodología proyecta la ruta que posibilitará al AGV desplazarse desde el punto de origen  $A$  hasta el punto de destino  $B$  en el lapso más corto posible, obviando la incidencia de las condiciones de tráfico.

En este sentido, es necesario abordar el desafío de la intersección entre las rutas de los AGVs. Para ello, se requiere desarrollar una lógica de paso que permita una gestión eficiente en dichas intersecciones. Por esta razón, se presentan los tres algoritmos más comúnmente utilizados (*STPRAN*, *STPFIFO* y *STPACC*), junto con el algoritmo adicional *STPCT*, considerado la solución más avanzada en este tipo de enfoques. En estos algoritmos, aunque se puede calcular la ruta manualmente, se utilizará la metodología expuesta en el Capítulo 3, la expuesta en el epígrafe 3.3.

Si consideramos que los vehículos utilizados cuentan diversos con sensores anticolidión que les permiten detenerse antes de colisionar con otros vehículos en el mismo tramo, la gestión del tráfico no será necesaria mientras circulen por un mismo corredor o pasillo. Sin embargo, cuando estos vehículos se encuentran en un cruce o intersección, pueden quedar detenidos indefinidamente. Por lo tanto, es crucial gestionar esta zona, a la que llamaremos *cruce* o *intersección*. En este punto, los Gestores de flotas se enfocarán en detener y permitir el paso de los AGV para evitar colisiones entre ellos. Los algoritmos convencionales propuestos difieren en su enfoque de gestión de estas intersecciones:

- ***Shortest Time Path with Random Priority Crossing (STPRAN)***:
  - Cuando un AGV va a entrar en una intersección, este algoritmo le permite el paso, si no se encuentra con ningún otro AGV dentro del cruce. Este tipo los cruces solo admiten un AGV circulando a la vez. Si, por lo contrario, se encuentra un AGV dentro del cruce, este no tendrá permiso de paso. La definición de un cruce se realiza manualmente.
  - Cuando varios AGV se encuentren esperando en diferentes entradas de la intersección, el AGV que procederá a entrar sigue un criterio aleatorio.
- ***Shortest Time Path with FIFO Priority Crossing (STPFIFO)***:
  - Como el algoritmo *STPRAN* permite solo un AGV a la vez en la intersección, cuando varios AGV se encuentren esperando por distintas entradas de la intersección, el AGV que procederá a entrar seguirá un criterio FIFO. La definición de cada uno de los cruces se realiza manualmente.
- ***Shortest Time Path with Accumulative Priority Crossing (STPACC)***:
  - Cuando un AGV va a entrar en una intersección, este algoritmo le permite el paso, si no se encuentra con ningún otro AGV dentro del cruce, o si el, o los, AGV que han entrado, lo han hecho por el mismo nodo de entrada. Este tipo los cruces puede admitir más de un AGV, si estos han entrado por el mismo nodo de entrada. Si, por lo contrario, se encuentra con algún otro AGV dentro del cruce y ha entrado por otra entrada, este no tendrá permiso de paso.

- Cuando varios AGV se encuentren esperando por diferentes entradas de la intersección, el AGV que procederá a entrar seguirá un criterio FIFO.
- Este algoritmo tiene el inconveniente de que puede priorizar una entrada de manera continuada, generando que otras entradas experimenten retrasos elevados. Es por ello que, para este algoritmo, limitaremos el número máximo de AGV que entran de manera continuada por un nodo de entrada.
- ***Shortest Time Path with Collision Table Crossing (STPCT)***:
  - Este algoritmo busca representar el máximo exponente de este tipo de algoritmos. Para ello, modelaremos el cruce definiendo todas las entradas y salidas, y relacionando todas sus conexiones mediante segmentos. Al ser requerida la entrada de un AGV por un segmento en concreto, si el gestor del cruce, según su modelo del entorno, considera que no va a colisionar con el resto de AGV que se encuentre dentro del cruce, le dejará paso. Si, por lo contrario, considera que podría causar una colisión, le denegará el paso.
  - Cuando varios AGV se encuentren esperando por diferentes entradas de la intersección, el AGV que procederá a entrar sigue un criterio FIFO.

## 5.4 Algoritmos no convencionales

En esta sección, se presentan algoritmos que aún no se utilizan comúnmente en la industria y cuyo origen se encuentra en la documentación científica. Estos algoritmos poseen el potencial de superar en rendimiento a los algoritmos convencionales, pero su uso no se ha extendido todavía en la industria por su dificultad de implementación. Con el objetivo de ilustrar este máximo potencial, se expondrán tres algoritmos de referencia específicamente diseñados para el tipo de entorno de la tesis. Se caracterizan por considerar el tráfico generado en la creación de las rutas. Los algoritmos seleccionados para esta tesis son los siguientes:

- ***Dynamic Geodesic Environment Partition (GEOPAR)***. Los algoritmos jerárquicos son aquellos que intentan combinar las mejores características de la distribución centralizada y descentralizada. Para representar uno de estos tipos de algoritmos propondremos la utilización del algoritmo expuesto en [37]. Este algoritmo se basa en la creación de dos capas de control. En la capa superior (la parte centralizada del algoritmo), se generan diferentes zonas. Al recibir una ruta que consiste en ir del punto  $A$ , que pertenece a la zona  $A'$ , al punto  $B$ , que pertenece a la zona  $B'$ , procederemos al cálculo de la secuencia de zonas por las que circulará el vehículo para ir desde el punto  $A$  hasta el punto  $B$ , teniendo en cuenta la distancia al centro de las diferentes zonas y el porcentaje de ocupación

de los AGV de cada zona. La capa inferior (la parte descentralizada del algoritmo) se encarga de la navegación dentro de estas zonas mediante un A\*.

- ***Conflict-free Multi-AGV Routing based on Improved A\* Algorithm (ASTAR)***. Para representar los algoritmos basados en el A\* aplicado a un grafo, proponemos el algoritmo citado en [20]. Se modificará el A\* estándar teniendo en cuenta el tiempo extra asociado al cálculo de las curvas que tomaría y la creación de ventanas de tiempo, las cuales se utilizan para ver el estado del tráfico de los nodos cercanos. Finalmente, el último punto relevante es que, al detectar que un AGV va a colisionar con otro AGV, el sistema de control recalcula la ruta de uno de los AGV para que no colisionen.
- ***Dynamic Routing of Automated Guided Vehicles in Real-time (TIMEWINDOW)***. Para representar los algoritmos basados en algoritmo *Dijkstra*, proponemos el algoritmo citado [13]. Este algoritmo busca incluir el tráfico en el cálculo de las rutas mediante la inclusión de ventanas de tiempo (donde se estima el tiempo de llegada de cada AGV a cada destino de las rutas mediante el cálculo *offline* del AGV). Por ello, al calcular la ruta de un AGV, tendremos en cuenta las posibles colisiones del resto de AGV mediante las ventanas de tiempo calculadas previamente.

## 5.5 Estado del arte en metodologías de aprendizaje

Para adaptar un Gestor de flotas a un entorno, se propone utilizar metodologías de aprendizaje. Existen muchos métodos de aprendizaje, cada uno cuenta con características y casos de aplicación diferentes. En esta tesis, se requiere una metodología que sea capaz de adaptar uno o varios algoritmos de Gestión de flotas para generar un Gestor de flotas adaptado al entorno.

Para ello, se necesita una metodología de aprendizaje que, dadas las entradas del entorno (el estado de los AGV y las tareas), adapte la respuesta del Gestor de flotas mediante las órdenes a los AGV, con el objetivo de minimizar los criterios de evaluación del entorno. Por lo tanto, la evaluación del aprendizaje se puede realizar mediante el resultado de los criterios de evaluación del entorno obtenidos mediante la simulación del entorno.

Dado que no se cuenta con un conjunto de datos de entrada y salida del sistema, se limita la metodología de aprendizaje. A continuación se presentan algunas metodologías susceptibles de ser utilizadas como metodología de adaptación al entorno.

El aprendizaje por refuerzo es un enfoque de aprendizaje automático en el que un agente aprende a tomar decisiones mediante la interacción con un

entorno, con el objetivo de maximizar una recompensa acumulativa a lo largo del tiempo. El agente no recibe ejemplos de entradas y salidas correctas, sino que aprende a través de la exploración y la retroalimentación recibida en forma de recompensas o castigos [41].

Los Procesos de Decisión de Markov (Markov Decision Process – MDP) son un marco matemático para modelar problemas de toma de decisiones secuenciales en entornos estocásticos. Los MDP son ampliamente utilizados en el aprendizaje por refuerzo para representar el entorno y la interacción entre el agente y el entorno [35].

Q-Learning es un algoritmo de aprendizaje por refuerzo basado en valores que busca aprender una función de valor, denominada función Q, que estima el valor esperado de la recompensa acumulativa futura para cada par acción-estado. El algoritmo utiliza la función Q para seleccionar acciones que sean capaces de maximizar la recompensa a largo plazo [42].

El Aprendizaje Profundo por Refuerzo (Deep Reinforcement Learning - DRL) combina técnicas de aprendizaje por refuerzo con redes neuronales profundas para aprender políticas de control complejas directamente a partir de datos en bruto, como imágenes o señales de sensores. Algunos ejemplos de algoritmos de DRL incluyen Deep Q-Networks (DQN) [34], Proximal Policy Optimization (PPO) [38] y Asynchronous Advantage Actor-Critic (A3C) [33].

Los algoritmos genéticos (GA) son una clase de algoritmos de optimización y búsqueda inspirados en la teoría de la evolución de Charles Darwin. Los GA funcionan mediante la evolución de una población de soluciones candidatas a lo largo de varias generaciones utilizando operadores de selección, cruce (crossover) y mutación. La calidad de las soluciones se evalúa mediante una función de aptitud (fitness), que guía el proceso de selección para favorecer a los individuos con mayor aptitud [14], [16], [31],[3],[24],[12] y [43].

Los algoritmos genéticos son especialmente útiles para abordar problemas de optimización difíciles y no convexos en los que las técnicas de búsqueda local y gradientes podrían quedar atrapadas en óptimos locales. Los GA son muy versátiles y se han aplicado en una amplia variedad de dominios, desde la optimización de funciones matemáticas hasta el diseño de estructuras y la planificación de rutas [32].

La programación genética [23] es un enfoque de algoritmo evolutivo que utiliza la evolución para optimizar programas de computadora. Los programas son representados como estructuras de árbol que expresan soluciones a un problema.

La programación genética utiliza operadores genéticos, como la cruza (crossover) y la mutación, para generar nuevas soluciones a partir de las existentes en una población. El proceso evolutivo continúa a lo largo de múltiples generaciones hasta que se alcanza una solución satisfactoria o un criterio de parada.

Las estrategias evolutivas [36] son una familia de algoritmos evolutivos que se originaron en la década de 1960 para resolver problemas de optimización numérica. Estos algoritmos evolutivos trabajan con representaciones numéricas de las soluciones y se enfocan en la adaptación de las estrategias de búsqueda en lugar de la explotación de la información específica del problema. Las estrategias evolutivas utilizan operadores genéticos como la mutación y la selección para generar nuevas soluciones y mejorar las existentes.

La optimización por enjambre de partículas [22] es un enfoque de optimización heurística inspirado en el comportamiento colectivo de organismos como enjambres de insectos o bandadas de aves. En este enfoque, cada partícula en el enjambre representa una solución potencial al problema de optimización y se mueve en el espacio de búsqueda siguiendo una combinación de su propia trayectoria, la posición óptima que ha encontrado y la posición óptima que el enjambre ha encontrado. La optimización por enjambre de partículas se ha aplicado con éxito a diversos problemas de optimización continua y discreta.

Estas diferentes metodologías de aprendizaje son susceptibles de ser utilizadas para adaptar un Gestor de flotas a un entorno. En el capítulo 8, se presentará la metodología de aprendizaje propuesta para esta tesis.

## Capítulo 6

# Entornos a evaluar

### Resumen

En este capítulo se presentan tres casos de uso basados en casos reales de la industria, que se utilizarán para evaluar los algoritmos expuestos en el Capítulo 5 y los algoritmos propuestos en esta tesis. Cada caso de uso se describe en detalle, incluyendo información sobre el espacio, los medios y los criterios de evaluación utilizados para cuantificar la reducción de costes. Adicionalmente, se presenta un entorno que será un caso de uso de laboratorio real, donde se realizaran las pruebas finales. Los apartados que se incluyen en este capítulo son los siguientes:

- **Introducción:** Se presenta una introducción que resume los diferentes casos de uso o entornos que se van a evaluar.
- **Entorno 1:** Se describe el primer entorno, incluyendo detalles sobre AGV y el método de reducción de costes.
- **Entorno 2:** Se describe el segundo entorno, incluyendo detalles sobre AGV y el método de reducción de costes.
- **Entorno 3:** Se describe el tercer entorno, incluyendo detalles sobre AGV y el método de reducción de costes.
- **Entorno Real:** Se describe el entorno de laboratorio real, incluyendo detalles sobre AGV y el método de reducción de costes.

## 6.1 Introducción

A continuación, mostraremos las principales características de los 4 entornos:

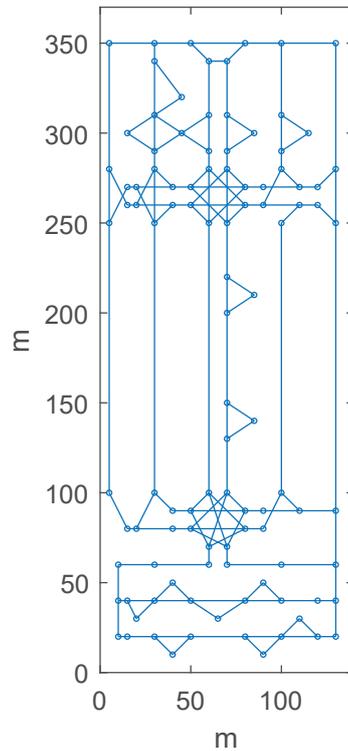
Características	Entorno 1	Entorno 2	Entorno 3	Entorno real
N.º nodos	104	477	649	54
N.º segmentos	151	595	830	65
N.º vehículos	140	30	30	2
N.º tareas	7	5	9	18
Distancia <i>Roadmap</i>	3675 m	439 m	2629 m	76 m
N.º destinos	14	105	62	3
Movilidad AGV	Uni*	Rot*	Bi*	Bi* Omni*
Distribución AGV	Cíclica	Única	Múltiple	Libre
Criterio	<i>Cri1</i>	<i>Cri2</i>	<i>Cri3</i>	<i>Cri4</i>
Indicador 1	<i>NMisCo</i>	<i>NAGV</i>	<i>NMisCo</i>	<i>NMisCo</i>
Indicador 2	<i>NAGV</i>	<i>TMedTobj</i>	<i>TMedTobj</i>	<i>NTmax</i>
Indicador 3	<i>TMedV</i>	<i>NTobj</i>	<i>NTobj</i>	<i>TMedV</i>
Duración	9 h	9 h	1 h 10 m	4 h
N.º de posibles rutas	Medio	Bajo	Alto	Bajo

Table 6.1.1: Tabla de características de los 4 entornos

- Bi\*: Bidireccional.
- Rot\*: Rotacional.
- Uni\*: Unidireccional.
- Omni: Omnidireccional.

A continuación se detallarán los 4 casos de uso utilizados en esta tesis, indicando para cada uno de ellos su mapa, sus AGV y su método de cuantificación.

## 6.2 Entorno 1



*Figura 6.2.1:* Entorno industrial número 1

En el primero de los entornos, se emular una línea de montaje secuencial donde las tareas de los AGV consistan en llevar material del almacén (que se encuentra en la parte inferior de la Figura 6.2.1) a diferentes partes de la línea de montaje (localizadas en la parte central y superior de la Figura 6.2.1).

Por la morfología del entorno, a la entrada y la salida del almacén se produce una elevada cantidad de tráfico. Esto ocurre porque el almacén solo cuenta una única salida y dos entradas, y una de las entradas comparte el mismo espacio con la salida; pasando, normalmente, las rutas más largas por la otra entrada al almacén. Es por este motivo, que una de las claves de este entorno consiste en la correcta elección de qué entrada tomar, según el volumen de tráfico, así como la adecuada gestión del tráfico de la intersección, que comparten tanto la salida como la entrada.



*Figura 6.2.2:* AGV modelo K10 de la empresa Kivnon, utilizado en el entorno 1

El número de AGV que se utilizan en este entorno no es un parámetro fijo, sino que contaremos con 20 AGV disponibles por cada tarea, siendo este valor superior al requerido por el Gestor de flotas. Se irán utilizando cuando el Gestor de flotas lo requiera. El objetivo de disponer de un número elevado de vehículos es poder alcanzar el número de tareas máximas que son capaces de realizar los vehículos con cada Gestor de flotas, y donde añadir un mayor número de AGV no aumenta el rendimiento. El número total de AGV disponibles será de 140 AGV.

Para este entorno, utilizaremos una distribución de AGV de rutas cíclicas. Los AGV esperarán en el almacén (en la posición de inicio de cada tarea) a recibir una misión. En ese mismo almacén será donde se colocarán los cargadores de las baterías.

### **6.2.1 AGV del entorno 1**

En este entorno, utilizaremos el modelo de AGV K10 de la empresa Kivnon, representado en la Figura 6.2.2. La movilidad de este vehículo es unidireccional, no pudiendo rotar ni cambiar de sentido, factor a tener en cuenta al calcular la ruta del AGV.

### **6.2.2 Cuantificación del entorno 1**

En este entorno nos centraremos en el primer caso de reducción de costes expuesto en la Sección 2.4, concretamente en el *Cri1*. El objetivo será maximizar el trabajo que pueden realizar los AGV en un entorno, independientemente del número de AGV utilizado. Para incrementar el trabajo que realizan los AGV, reduciremos el tiempo de entrega del material de cada una de las tareas mediante la variable  $Te_n$ . Todas las tareas tendrán el mismo valor.

La extracción del trabajo máximo que son capaces de realizar los AGV la

haremos usando el valor máximo obtenido  $NMisCo$ , cumpliendo con todas las misiones solicitadas entre todos los valores de  $Te$ .

Otros parámetros secundarios para cuantificar el rendimiento de los algoritmos en cada carga de trabajo de los AGV serán los siguientes:

- $NAGV$ . Número de AGV que han sido utilizados en un entorno para cumplir las tareas y requisitos de este.
- $TMedV$ . Tiempo medio de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.

### 6.3 Entorno 2

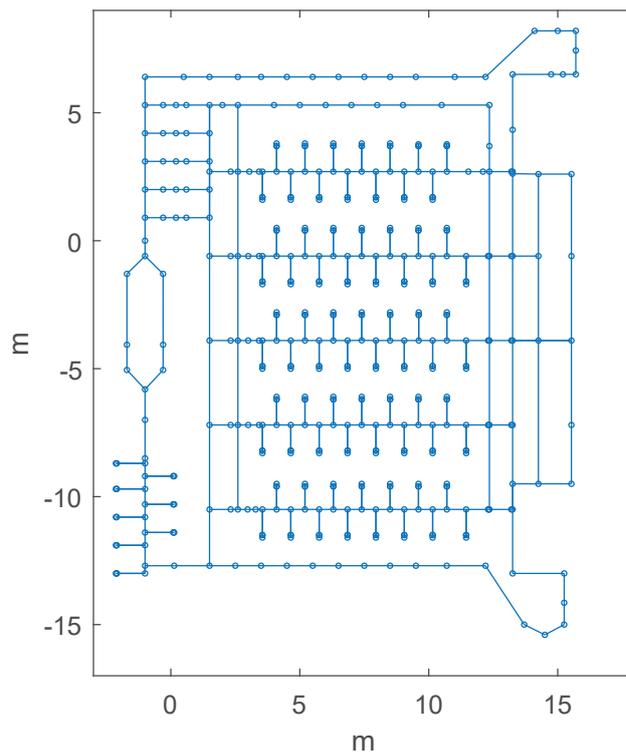


Figura 6.3.1: Entorno industrial número 2

En el caso del segundo entorno planteado, proponemos un caso industrial real donde los AGV mueven el material de producción entre diferentes puntos de trabajo y un almacén. Realizarán tareas de transporte a lugares donde el vehículo se detiene durante un período corto de tiempo, realizando funciones de línea de montaje o de gestión del material en el almacén.

El *Roadmap* para este entorno se encuentra descrito en la Figura 6.3.1. Como se puede observar, el mapa es de un tamaño reducido, haciendo que el número de posibles rutas para ir de un destino a otro sea bajo, ya que en muchas ocasiones solo existe una ruta posible para hacer ese trayecto. Por otra parte, al ser un entorno tan reducido con muchas vías entrelazadas, genera una gran cantidad de cruces, todos ellos sencillos. Cabe señalar que todos los caminos son de un único sentido.

En este entorno, la distribución de los vehículos se gestionará desde un punto de salida único de los AGV, donde recibirán las misiones que, a su vez, tendrán como último destino, de nuevo, el punto de salida único.

La carga de baterías de los vehículos se realizará mediante una zona de carga de baterías, justo antes de llegar al punto de salida único, donde los vehículos cargarán sus baterías cuando estas estén descargadas.

### 6.3.1 AGV del entorno 2



*Figura 6.3.2:* AGV modelo K05 de la empresa Kivnon, utilizado en el entorno 2

En este entorno, utilizaremos el modelo de AGV K05 de la empresa Kivnon, representado en la Figura 6.3.2. La movilidad de este vehículo es rotacional, teniendo que rotar para cambiar de sentido o girar en zonas estrechas. Este es un factor a tener en cuenta al calcular la ruta del AGV.

### 6.3.2 Cuantificación del entorno 2

En este entorno, nos centraremos en el segundo caso de reducción de costes expuesto anteriormente en la Sección 2.4, concretamente en el *Cri2*; donde se pretende minimizar el número de AGV necesarios para realizar un trabajo específico. Por lo que el número de tareas y el tiempo  $Te$  de las mismas será fijo. Lo que variará será el número de AGV disponibles.

La evaluación de la cantidad de AGV necesarios se hará mediante la variable  $NAGV$ , siendo el algoritmo que obtenga un valor mínimo de  $NAGV$  el que obtenga mayor rendimiento. Para extraer el valor mínimo, se partirá de un número alto de AGV disponibles, que se irá reduciendo hasta que el algoritmo no sea capaz de completar todas las misiones dentro del tiempo estipulado.

Otros parámetros secundarios para cuantificar el rendimiento de los algoritmos cuando obtengan el mismo valor de  $NAGV$  serán los siguientes:

- $TMedT_{obj}$ . Tiempo medio de llegada de los AGV a sus destinos respecto a sus  $T_{obj}$  respectivos. Medido en segundos.
- $NT_{obj}$ . Número de veces que los AGV han llegado a sus destinos después del  $T_{obj}$ .

El  $T_{obj}$  para cada destino y tarea  $T_{obj_{Task, Dest}}$  se calcula a partir del algoritmo de cálculo ideal  $STPI$  mediante el parámetro  $TSinTra_{Task, Dest}$  (Tiempo de desplazamiento desde el destino de inicio de la tarea  $Task$  hasta el destino  $Dest$  de esa misma tarea, sin considerar el tráfico). Lo calcularemos mediante la siguiente fórmula:

$$T_{obj_{Task, Dest}} = TSinTra_{Task, Dest} * 1.2 \quad (6.3.1)$$

### 6.4 Entorno 3

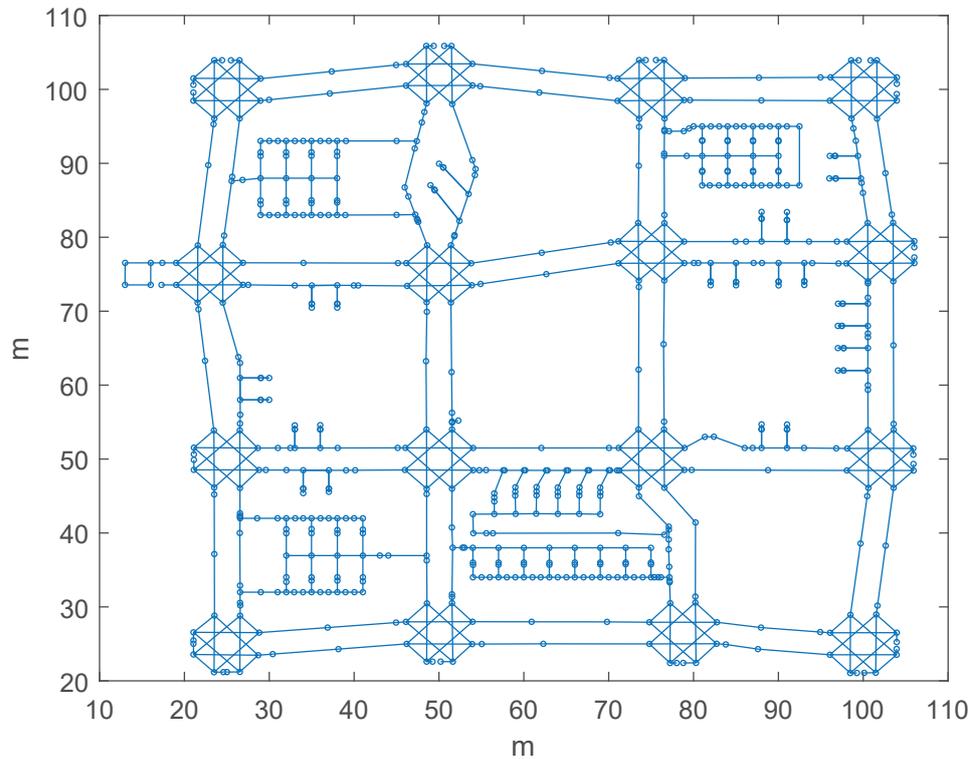
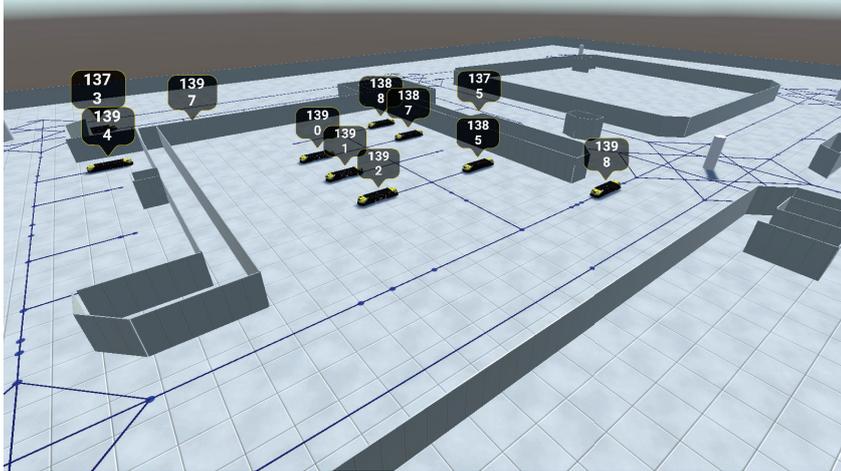


Figura 6.4.1: Entorno industrial número 3

Para el tercer entorno, representaremos una instalación más compleja basada en instalaciones reales del sector de la automoción. El entorno estará separado por islas, que podrán ser: almacenes o líneas de montaje de elementos secundarios o principales. Los AGV tendrán que realizar tarea diversas mediante el movimiento de mercancías por el interior de la fábrica.

Estas islas suelen estar rodeadas por pasillos en los que son capaces de circular entre uno y dos vehículos en paralelo. En este caso, los pasillos serán bidireccionales, ya que este es el caso más complejo, generando así muchas rutas posibles. Para este entorno, utilizaremos nueve islas irregulares, como se puede ver en la Figura 6.4.1, donde se encuentran puntos de recogida y entrega repartidos por el entorno. Estos puntos de entrega tienen múltiples posiciones de destino, que se van ocupando y desocupando con los vehículos que van cumpliendo



*Figura 6.4.2:* Visualización del entorno industrial con un motor gráfico en 3D

las diferentes tareas. Para este ejemplo, utilizaremos estos tres tipos de entrega con múltiples estaciones, pudiendo tener 2, 6 u 8 estaciones que cuentan con diferentes tipologías y número de entradas y salidas.

Con el objetivo de representar todo tipo de tareas distintas que realizar, en el entorno se utilizarán un total de 17 puntos de entrega con múltiples estaciones, siendo estos de los tres tipos ya mencionados, concretamente los que se pueden ver en la Figura 6.4.3. Estos puntos de entrega están programados para que tengan una gestión propia, en la que deciden la estación apropiada y gestionan el tráfico dentro de los puntos de entrega. El objetivo de utilizar puntos de entrega múltiples se debe a que es habitual que una tarea requiera ser realizada por varios AGV, siendo, a su vez, necesario que los puntos de entrega tengan varias estaciones para poder acomodar varios vehículos en los lugares de destino. Otro motivo para incluir estos puntos de entrega múltiple es que en ellos se genera un tráfico que repercute en el sistema.

En este entorno, hay una variedad de caminos que pueden ser utilizados para llegar a los diferentes destinos. La elección de qué camino tomar es un factor relevante. Por ello, los algoritmos que mejor seleccionen las rutas, siempre teniendo en cuenta el tráfico, obtendrán un mejor rendimiento. Por otro lado, al tener puntos de entrega entrelazados, se genera mucho tráfico cruzado debido a que los AGV entrecruzan sus rutas. En este contexto, la gestión de las intersecciones es muy relevante.

En este entorno, utilizaremos una distribución de los AGV con múltiples

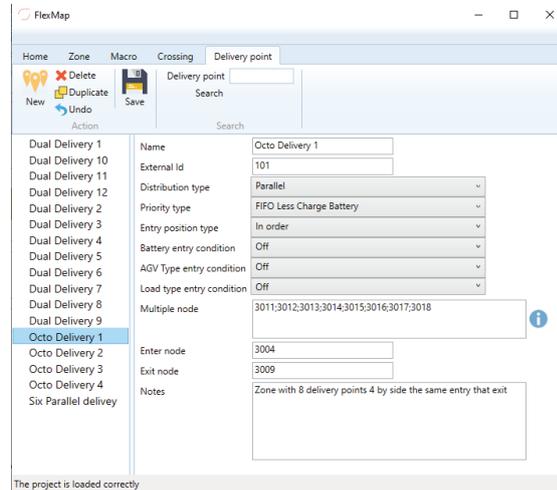


Figura 6.4.3: Definición de los puntos de entrega

puntos de descanso. Estos puntos serán los 17 anteriormente mencionados. La carga de baterías se realizará por oportunidad en en estos 17 puntos de descanso.

### 6.4.1 AGV del entorno 3



Figura 6.4.4: AGV modelo K11 de la empresa Kivnon, utilizado en el entorno 3

En este entorno, utilizaremos el modelo de AGV K11 de la empresa Kivnon, que se puede ver en la Figura 6.4.4. La movilidad de este vehículo es bidireccional y puede cambiar de sentido cuando sea necesario. Por el contrario, este modelo no es capaz de rotar.

### 6.4.2 Cuantificación del entorno 3

Este entorno se centrará en el tercer caso de reducción de costes expuesto anteriormente en la Sección 2.4, concretamente el *Cri3*. En este caso, pretendemos

estudiar cómo maximizar el trabajo que pueden realizar los AGV en un entorno con un número fijo de AGV.

El indicador de rendimiento en este entorno será la variable  $NMisCo$ , en la que se irán modificando el número total de misiones a completar mediante la variación de los tiempos entre peticiones  $Te$  de las tareas. Como este valor de  $Te$  es diferente para cada tarea, para simplificar la lectura de los resultados y mejorar la visualización, en vez de analizar el impacto en el rendimiento de la gestión de los algoritmos con diferentes valores de  $Te$  para cada tarea, se mostrará este valor como porcentaje respecto al valor obtenido mediante el algoritmo  $STPI$ , expuesto en la Sección 3.3. Por ello, un valor de 100 % significa el nivel máximo de rendimiento que idealmente se podría gestionar. El nivel de tráfico y de estrés del sistema se reducirá cuando se reduzca este valor porcentual, al que hemos denominado  $PST$ . El valor de  $Te_n$  de cada tarea se obtiene mediante la siguiente fórmula:

$$Te_{Task} = \frac{TTTare_{Task} * 100}{NAGVDis_{Task} * PST} \quad (6.4.1)$$

Donde:

- $TTTare_{Task}$ : Tiempo de desplazamiento total necesario para completar una tarea, sin considerar el tráfico.
- $PST$ : Porcentaje de carga de trabajo del  $Te$  respecto a tiempo de desplazamiento  $TTTare_{Task}$ .
- $NAGVDis_{Task}$ : Número de AGV asignados a una tarea.

Solo se comparan entre sí los resultados con el mismo valor de  $PST$ . Empezaremos con valores de carga de trabajo bajos, partiendo de un 10 % de  $PST$ , e incrementándolo a razón del 10 %, hasta que los AGV no sean capaces de completar todas las tareas. Con un valor máximo de 100 % de  $PST$ .

Otros parámetros secundarios para cuantificar el rendimiento de los algoritmos para cada porcentaje de  $PST$  serán los siguientes:

- $N_{Tobj}$ . Número de veces que los AGV han llegado a sus destinos después del  $Tobj$ .
- $TMedTobj$ . Tiempo medio de llegada de los AGV a sus destinos respecto a sus  $Tobj$  respectivos. Medido en segundos.

El tiempo  $Tobj$  y  $Tmax$  para cada destino y tarea mediante las variables  $Tobj_{Task, Dest}$  y  $Tmax_{Task, Dest}$  lo asignaremos de manera proporcional al tiempo de viaje extraído del algoritmo  $STPI$ , generando  $TSinTraf_{Task, Dest}$

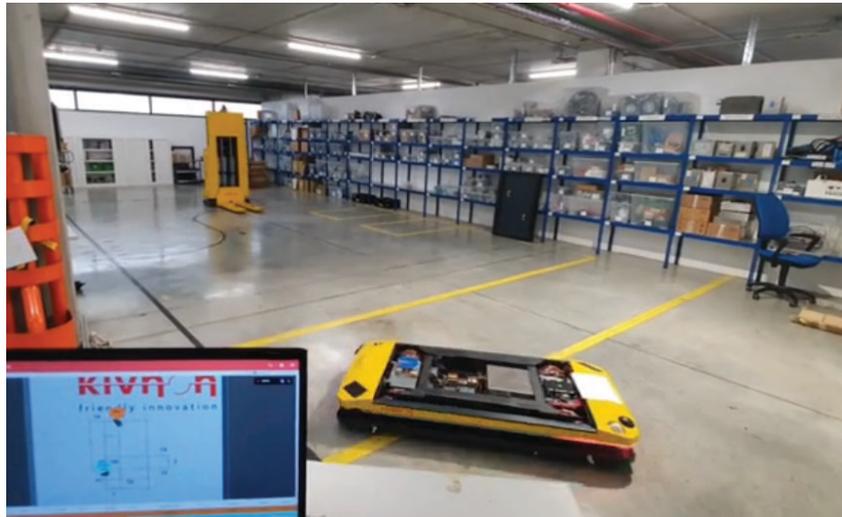
mediante la siguiente fórmula:

$$T_{objTask, Dest} = T_{SinTrafTask, Dest} * 1.2 \quad (6.4.2)$$

$$T_{maxTask, Dest} = T_{SinTrafTask, Dest} * 1.4 \quad (6.4.3)$$

Consideraremos que un algoritmo no ha sido capaz de completar una tarea cuando alguno de los AGV no haya sido capaz de llegar dentro del tiempo  $T_{max}$  a cualquiera de sus destinos.

## 6.5 Entorno Real



*Figura 6.5.1:* Imagen tomada durante las pruebas en un entorno de laboratorio real

Finalmente, se presenta un entorno real para la realización de una prueba del algoritmo propuesto en un entorno real. En este caso, el entorno real es el laboratorio de I+D+i de Kivnon, donde se ha habilitado una zona para la realización de las pruebas con 2 AGV que se moverán realizando tareas definidas para esta prueba.

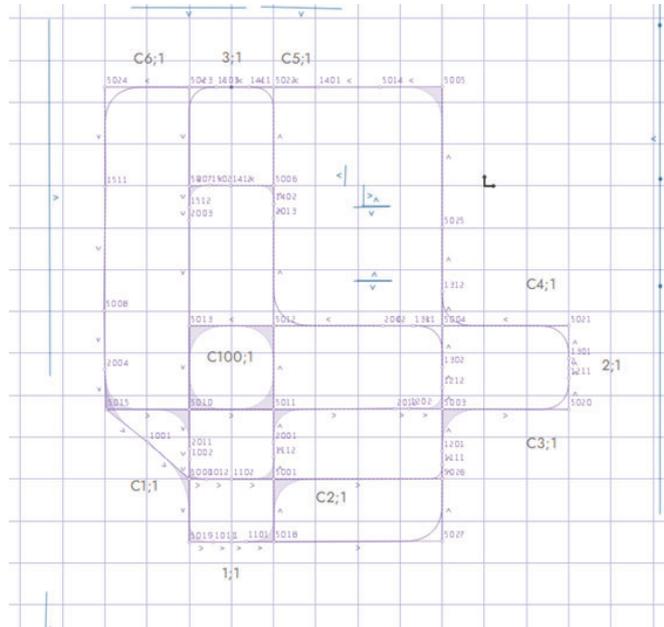
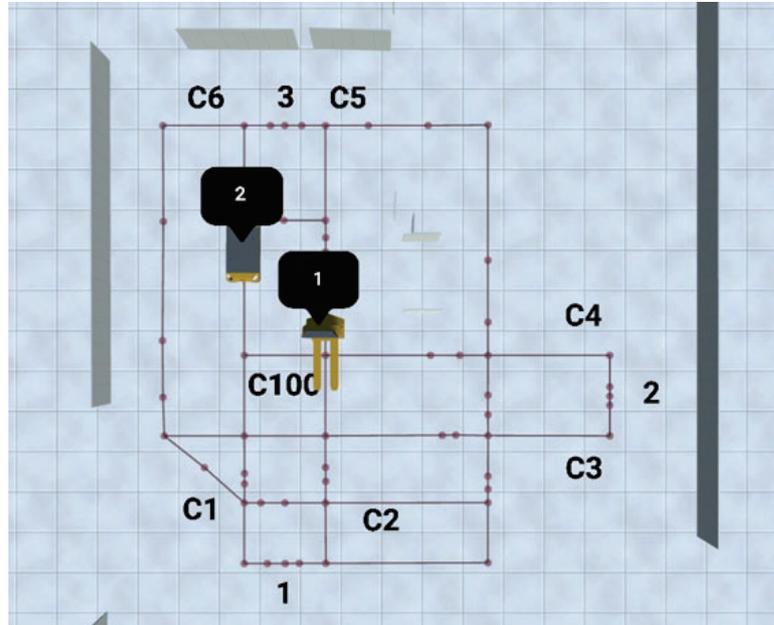


Figura 6.5.2: Entorno de laboratorio real

El entorno seleccionado se ilustra en la Figura 6.5.1. Dado que este entorno no tiene las dimensiones típicas de las instalaciones, se diseñó específicamente el *Roadmap* mostrado en la Figura 6.5.2, con el objetivo de proporcionar diferentes rutas posibles hacia distintos destinos. Generando que la ruta más rápida sin tráfico es aquella que conlleva más tráfico en este contexto. Además, se cuenta con la representación en 3D en tiempo real de los AGV en este entorno, como se puede apreciar en la Figura 6.5.3.



*Figura 6.5.3:* Visualización del entorno de laboratorio real con un motor gráfico en 3D

### 6.5.1 AGV del entorno real



*Figura 6.5.4:* AGV modelo K41 de la empresa Kivnon, utilizado en el entorno real

En el siguiente entorno de laboratorio, se utilizarán los modelos de AGV K41 y K55, desarrollados por la empresa Kivnon. Dichos modelos pueden apreciarse en las Figuras 6.5.4 y 6.5.5, respectivamente. El modelo K51 posee movilidad bidireccional, lo que le permite cambiar de sentido cuando sea requerido; sin embargo, no cuenta con la capacidad de rotación. Por otro lado, el modelo K41 goza de movilidad omnidireccional y también puede cambiar de sentido cuando sea necesario. La elección de estos modelos se debe a que eran los disponibles para su utilización.



*Figura 6.5.5:* AGV modelo K55 de la empresa Kivnon, utilizado en el entorno real

### **6.5.2 Cuantificación del entorno real**

Dadas las dimensiones de la instalación y los recursos disponibles, se contemplan 3 posibles posiciones de entrega y 2 AGV. Con el objetivo de maximizar el tráfico, se generan peticiones a dos tareas de manera simultánea. El trabajo llevado a cabo en este entorno busca emular el comportamiento más desfavorable en una instalación. Al analizar las posibles tareas para estos 3 destinos de forma simultánea, que solo cuenten con dos destinos, se obtienen las mostradas en la Figura 6.5.6.

ID	Inicio Pos1	Inicio Pos2	Inicio Pos3	Llegada Pos1	Llegada Pos2	Llegada Pos3
1	AGV 1	AGV 2	Libre	AGV 2	AGV 1	Libre
2	AGV 1	AGV 2	Libre	AGV 2	Libre	AGV 1
3	AGV 1	AGV 2	Libre	Libre	AGV 1	AGV 2
4	AGV 2	AGV 1	Libre	AGV 1	AGV 2	Libre
5	AGV 2	AGV 1	Libre	AGV 1	Libre	AGV 2
6	AGV 2	AGV 1	Libre	Libre	AGV 2	AGV 1
7	AGV 1	Libre	AGV 2	AGV 2	AGV 1	Libre
8	AGV 1	Libre	AGV 2	AGV 2	Libre	AGV 1
9	AGV 1	Libre	AGV 2	Libre	AGV 2	AGV 1
10	AGV 2	Libre	AGV 1	AGV 1	AGV 2	Libre
11	AGV 2	Libre	AGV 1	AGV 1	Libre	AGV 2
12	AGV 2	Libre	AGV 1	Libre	AGV 1	AGV 2
13	Libre	AGV 1	AGV 2	AGV 1	AGV 2	Libre
14	Libre	AGV 1	AGV 2	Libre	AGV 2	AGV 1
15	Libre	AGV 1	AGV 2	AGV 2	Libre	AGV 1
16	Libre	AGV 2	AGV 1	AGV 2	AGV 1	Libre
17	Libre	AGV 2	AGV 1	Libre	AGV 1	AGV 2
18	Libre	AGV 2	AGV 1	AGV 1	Libre	AGV 2

Figura 6.5.6: Posibles tareas para el entorno real

Existen un total de 18 combinaciones posibles; no obstante, al omitir aquellas en las que los AGV están invertidos, el total de combinaciones se reduce a 9. Por lo tanto, si generamos una secuencia, las tareas a realizar en este entorno serán las que se indican en la Figura 6.5.7.

Secuencia por ID	Conjunto de tareas 1			Conjunto de tareas 2		
	Origen	Destino	ID tarea	Origen	Destino	ID tarea
1	Pos1	Pos2	T1_1-2	Pos2	Pos1	T1_2-1
5	Pos2	Pos1	T2_2-1	Pos1	Pos3	T2_1-3
8	Pos1	Pos3	T3_1-3	Pos3	Pos1	T3_3-1
12	Pos3	Pos2	T4_3-2	Pos1	Pos3	T4_1-3
15	Pos2	Pos3	T5_2-3	Pos3	Pos1	T5_3-1
7	Pos3	Pos1	T6_3-1	Pos1	Pos2	T6_1-2
6	Pos1	Pos2	T7_1-2	Pos2	Pos3	T7_2-3
17	Pos2	Pos3	T8_2-3	Pos3	Pos2	T8_3-2
13	Pos3	Pos2	T9_3-2	Pos2	Pos1	T9_2-1

Figura 6.5.7: Tareas que forman parte del entorno real

Por lo que se irán ejecutando el conjunto de tareas 1 y 2, con peticiones cada 100 segundos, esta secuencia completa de combinaciones se repetiría durante un periodo de 4h. Para cada uno de los destinos de las tareas, se le aplica un  $T_{max}$ .

El criterio de evaluación para este entorno será el  $Cri4$ . Por lo tanto, se realizará la evaluación del rendimiento con una única condición de trabajo y

unos AGV específicos. Los indicadores que se utilizarán para su evaluación, en orden de importancia, son los siguientes:

1.  $NMisCo$ . Número de misiones completadas en un entorno, cumpliendo los requisitos del mismo, en un periodo de tiempo específico.
2.  $NotNMisCo$ . Número de misiones no completadas en un entorno, en un periodo de tiempo específico.
3.  $NT_{max}$ . Número de veces que los AGV han llegado a sus destinos después del  $T_{max}$ .
4.  $TMedT_{max}$ . Tiempo medio de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
5.  $TMaxT_{max}$ . Tiempo máximo de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
6.  $TMinT_{max}$ . Tiempo mínimo de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
7.  $TMedV$ . Tiempo medio de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.
8.  $TMaxV$ . Tiempo máximo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.
9.  $TMinV$ . Tiempo mínimo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.



## Capítulo 7

# Análisis de rendimiento de los algoritmos de referencia

### Resumen

Para analizar el rendimiento de los algoritmos de referencia expuestos en el Capítulo 5, se procederá a la evaluación de estos algoritmos en los 3 primeros entornos del Capítulo 6. En este capítulo se presentarán los resultados obtenidos, los cuales serán analizados con los criterios de evaluación y la carga de computación de cada entorno. Finalmente, se realizará una comparativa de los algoritmos, donde se analizarán sus ventajas y desventajas. De estos resultados también se extraerá cuáles son los aspectos que tienen mayor potencial de mejora.

Este capítulo está estructurado en los siguientes apartados:

- **Rendimiento de los algoritmos de Gestión de flotas en cada entorno:** En este apartado se cuantificará la reducción de costes conseguida por cada algoritmo en cada uno de los entornos evaluados. Se compararán los resultados obtenidos por cada algoritmo para determinar cuál de ellos ofrece el mejor rendimiento en términos de reducción de costes.
- **Carga computacional de los algoritmos de Gestión de flotas en cada entorno:** En este apartado se analizará la carga de trabajo requerida por cada algoritmo en cada entorno, con el objetivo de evaluar su eficiencia en

términos de recursos computacionales. Se medirá el tiempo de ejecución necesario para cada algoritmo, con el fin de determinar cuál de ellos es más eficiente en términos de uso de recursos computacionales.

- **Análisis de los algoritmos de Gestión de flotas:** En este apartado se realizará un análisis detallado del rendimiento de cada algoritmo, con el fin de identificar las condiciones de trabajo donde cada algoritmo presenta un mayor rendimiento. Se identificarán las fortalezas y debilidades de cada algoritmo para determinar en qué condiciones es más efectivo cada uno.
- **Comparativa del rendimiento de los diferentes algoritmos de referencia:** En este apartado se comparará el rendimiento de los diferentes algoritmos de referencia entre sí, extrayendo las diferencias de rendimiento entre ellos en las diferentes situaciones analizadas.
- **La mejor combinación:** En este apartado se identificarán las mejores características de cada algoritmo con el objetivo de utilizarlas en la creación de un nuevo algoritmo de Gestión de flotas. Se evaluarán las combinaciones posibles de características de los algoritmos existentes para crear un nuevo algoritmo que ofrezca un rendimiento superior en términos de reducción de costes y eficiencia computacional.

## 7.1 Rendimiento de los algoritmos de Gestión de flotas en cada entorno

En este apartado, procederemos a recoger los resultados de los 3 primeros entornos expuestos en el Capítulo 6.

### 7.1.1 Entorno 1

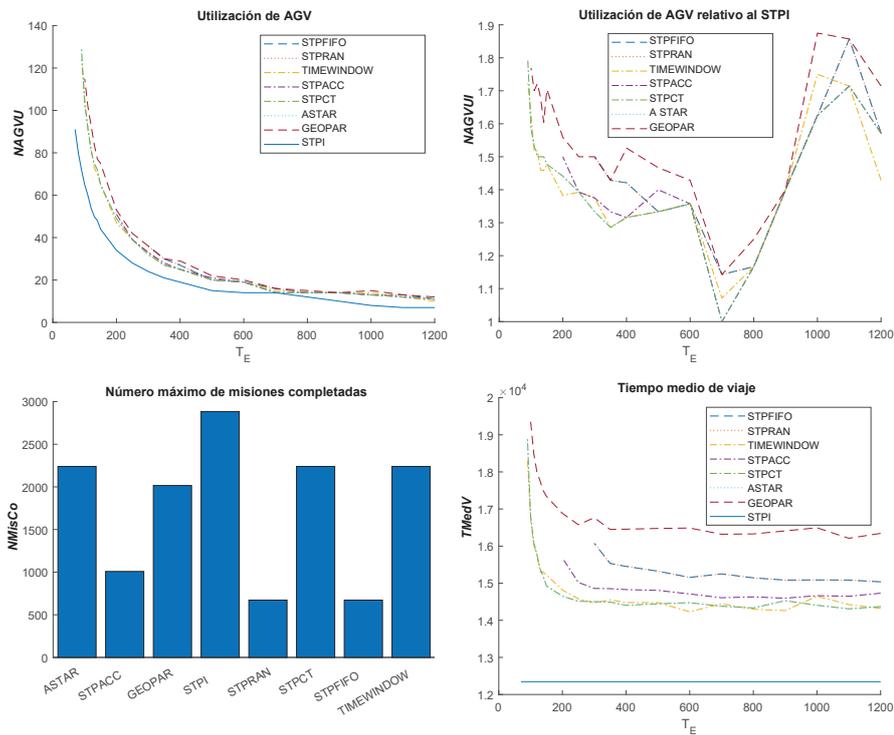


Figura 7.1.1: Resultado de los algoritmos de referencia del entorno 1

En este apartado, evaluaremos el rendimiento de los algoritmos expuestos en el Capítulo 5 en el entorno 1 de la Sección 6.2. Utilizaremos el criterio de evaluación del entorno 1 y expondremos los resultados de estos criterios en la Figura 7.1.1. Los parámetros expuestos en las gráficas de la figura son los siguientes:

- **Gráfica superior izquierda:**
  - **Eje X:**  $T_e$  (Periodo entre peticiones de una tarea. Medido en segundos).

- **Eje Y:** *NAGV* (Número de AGV que han sido utilizados en un entorno para cumplir las tareas y requisitos de este).
- **Valoración del eje Y:** Valor más bajo mejor.
- **Gráfica superior derecha:**
  - **Eje X:** *Te* (Periodo entre peticiones de una tarea. Medido en segundos).
  - **Eje Y:** *NAGVUI* (Valor de *NAGV* respecto al valor ideal. Este se obtiene mediante el algoritmo *STPI*).
  - **Valoración del eje Y:** Valor más bajo mejor.
- **Gráfica inferior izquierda:**
  - **Eje X:** Algoritmos de Gestión de flotas.
  - **Eje Y:** *NMisCo* (Número de misiones completadas en un entorno, cumpliendo los requisitos del mismo, en un periodo de tiempo específico).
  - **Valoración del eje Y:** Valor más alto mejor.
- **Gráfica inferior derecha:**
  - **Eje X:** *Te* (Periodo entre peticiones de una tarea. Medido en segundos).
  - **Eje Y:** *TMedV* (Tiempo medio de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos).
  - **Valoración del eje Y:** Valor más bajo mejor.

El parámetro más relevante para evaluar el rendimiento de estos algoritmos en este entorno es el número máximo de misiones completadas satisfactoriamente por cada uno de estos algoritmos sin que ninguna de las misiones se queden sin completar. Este parámetro se encuentra en la gráfica inferior izquierda de la Figura 7.1.1 y se visualiza mediante la variable *NMisCo*.

El segundo parámetro más relevante es el número de AGV utilizados para cada rango de *Te*, indicado mediante la variable *NAGV*. Representa el mismo valor respecto al algoritmo *STPI*, mediante la variable *NAGVUI*. Estos dos parámetros se encuentran cuantificados en las dos gráficas superiores de la Figura 7.1.1.

El último parámetro para evaluar estos algoritmos es el tiempo medio de los trayectos de los AGV, mediante la variable *TMedV* que aparece en la gráfica inferior derecha de la Figura 7.1.1.

Si nos centramos en el criterio más relevante (el número de misiones máximas completadas *NMisCo*), siendo este un valor único utilizado para evaluar el rendimiento máximo sin tener en cuenta el número de AGV usados, el orden creciente de rendimiento de los algoritmos según este parámetro será el siguiente:

1. El *STPRAN* y el *STPFIFO* son los dos algoritmos con menor rendimiento. Si analizamos la causa de este bajo rendimiento, podemos deducir que la generación de rutas fijas no es el factor determinante de su bajo rendimiento, sino que este está causado por su gestión de las intersecciones, que genera mucho tráfico en los cruces, al permitir el paso de un solo AGV a la vez. Sobre todo, este tráfico se genera a la entrada y salida del almacén.
2. El *STPACC* tiene un rendimiento ligeramente superior a los dos anteriores. Son algoritmos muy similares, pero este último, al contar con una gestión de intersecciones que permite un mayor paso de AGV en las intersecciones, facilita que se genere un mayor flujo de paso de AGV en la entrada y salida del almacén, obteniendo un rendimiento ligeramente superior.
3. El *GEOPAR* sale claramente perjudicado en este entorno. Esto es debido a que este entorno tiene muchas rutas unidireccionales, lo cual genera que el *Top Layer* no tenga en cuenta todas sus peculiaridades y causa que los AGV, al tener que cruzar por las *Top Layer*, sigan rutas que dan muchos rodeos y que no siempre son las más óptimas.
4. El algoritmo *STPCT* se ve claramente beneficiado en este entorno respecto a otros algoritmos, ya que el número de rutas disponibles es limitado. Por lo que, al utilizar rutas fijas, no se ve perjudicado, y al contar con una buena gestión de los cruces, obtiene un buen resultado que en otros tipos de mapas no tendría.
4. El *ASTAR* no se ve perjudicado en este entorno, ya que el entorno no desfavorece el cálculo mediante  $A^*$ . En otro tipo de entornos en los que la ruta más óptima no sea la más cercana, su rendimiento puede verse afectado. El buen rendimiento en este entorno se debe a la buena gestión en los cruces.
4. El *TIMWINDOW* en este mapa no explota demasiado el cálculo de rutas que tienen en cuenta el tráfico (mediante las ventanas de tiempo), ya que, normalmente, la ruta más óptima con tráfico es la misma que sin tráfico. Sin embargo, con mucho tráfico logra un rendimiento ligeramente superior. Aun sin contar con esta ventaja, en este entorno el algoritmo *TIMWINDOW* obtiene un buen rendimiento.

En la evaluación anterior hemos utilizado un único criterio: el *NMisCo*, siendo este valor muy similar en varios algoritmos. Por este motivo, vamos a utilizar los siguientes criterios de evaluación: *NAGV*, *NAGVUI* y *TMedV*, con el objetivo de discernir las diferencias de rendimiento.

En este entorno, al contar con una distribución de los AGV con una ruta cíclica, el número de AGV utilizado puede variar en gran medida con una pequeña diferencia de rendimiento en el tiempo de desplazamiento, sobre todo con poca carga de trabajo. Es posible que una tarea haya utilizado solo un AGV durante casi todo el tiempo y que, una vez, haya tenido que utilizar un AGV extra

porque uno de los vehículos ha tardado unos pocos segundos más en llegar al destino. Es decir, que al acumular unos segundos de retraso, pueden observarse grandes diferencias de rendimiento.

Por ese motivo, si evaluamos *NAGV* y *TMedV*, podemos extraer bastantes conclusiones. Si ordenamos el rendimiento según *TMedV* de manera creciente, el orden resultante será el siguiente:

1. El *GEOPAR*, como se ha mencionado anteriormente, al ser un entorno irregular con caminos unidireccionales, genera que algunas de las rutas sean más largas al tener que circular por los *top layer*, lo cual genera, a su vez, que los vehículos den rodeos. Pero aun contando con un mayor tiempo de circulación que los demás algoritmos, gracias a su gestión del tráfico es capaz de conseguir un mayor número de tareas completadas con mayor tráfico, que los otros 3 algoritmos con menos rendimiento. Esto lo logra a costa de utilizar un mayor número de AGV.
2. En el *STPRAN* y el *STPFIFO* podemos observar que el tiempo medio de trayecto es elevado, lo cual genera una mayor utilización de AGV y da como resultado un rendimiento bajo.
3. Al analizar el tiempo medio de circulación del *STPACC*, observamos que tiene un valor inferior que sus algoritmos semejantes. El motivo por el que genera un mejor rendimiento que el *STPRAN* y el *STPFIFO* es por las diferencias en la gestión de las intercesiones.
4. Respecto a los otros 3 algoritmos: el *STPCT*, el *TIMEWINDOW* y el *ASTAR*, las diferencias, teniendo en cuenta estos criterios, es baja, no pudiendo deducir conclusiones de rendimientos entre ellos. Con la salvedad de que, con mucho tráfico, el *TIMEWINDOW* es capaz de extraer un rendimiento marginalmente superior.

### 7.1.2 Entorno 2

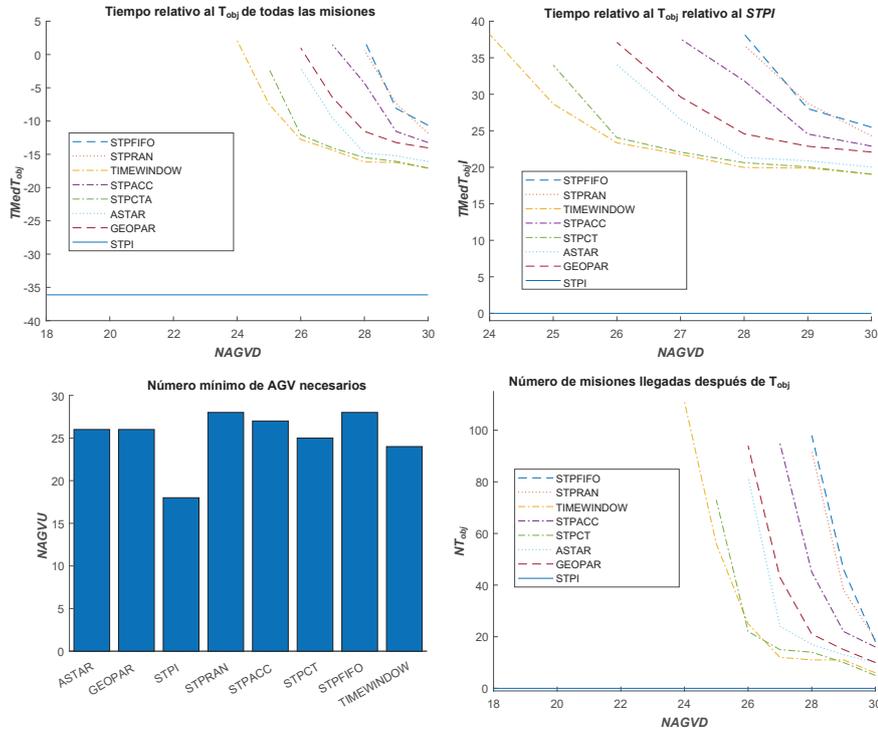


Figura 7.1.2: Resultado de los algoritmos de referencia del entorno 2

El segundo entorno propuesto en la sección 6.3 es bastante diferente al anterior, ya que es mucho más pequeño, cuenta con pocas rutas alternativas para llegar a su destino y con muchos cruces simples muy juntos. Todo esto provoca que en este entorno lo más relevante sea la gestión de los cruces y la interacción entre ellos. Utilizaremos el criterio de evaluación del entorno 2 y expondremos los resultados de estos criterios en la Figura 7.1.2. Los parámetros expuestos en las gráficas de la figura son los siguientes:

- **Gráfica superior izquierda:**
  - **Eje X:**  $NAGVD$  (Número de AGV disponibles para ser utilizados en un entorno).
  - **Eje Y:**  $T_{MedT_{obj}}$  (Tiempo medio de llegada de los AGV a sus destinos respecto a sus  $T_{obj}$  respectivos. Medido en segundos).
  - **Valoración del eje Y:** Valor más bajo mejor.
- **Gráfica superior derecha:**
  - **Eje X:**  $NAGVD$  (Número de AGV disponibles para ser utilizados en

- un entorno).
- **Eje Y:**  $TMedT_{obj}I$  (Valor de  $TMedT_{obj}$  respecto al obtenido por el algoritmo  $STPI$ . Medido en segundos).
- **Valoración del eje Y:** Valor más bajo mejor.
- **Gráfica inferior izquierda:**
  - **Eje X:** Algoritmos de referencia.
  - **Eje Y:**  $NAGV$  (Número de AGV que han sido utilizados en un entorno para cumplir las tareas y requisitos de este).
  - **Valoración del eje Y:** Valor más alto mejor.
- **Gráfica inferior derecha:**
  - **Eje X:**  $NAGVD$  (Número de AGV disponibles para ser utilizados en un entorno).
  - **Eje Y:**  $NT_{obj}$  (Número de veces que los AGV han llegado a sus destinos después del  $T_{obj}$ ).
  - **Valoración del eje Y:** Valor más bajo mejor.

El criterio más relevante en este entorno es el número mínimo de AGV necesarios para realizar las tareas en un entorno. Para ello, se utiliza el valor  $NAGVD$  mínimo que el Gestor de flotas ha sido capaz de gestionar. Este valor se puede encontrar expuesto en la gráfica inferior izquierda de la Figura 7.1.2.

Adicionalmente, como parámetros complementarios para la evaluación, se encuentran el tiempo medio relativo al tiempo objetivo  $T_{obj}$  mediante la variable  $TMedT_{obj}$ , y el mismo valor relativo al tiempo sin tráfico  $STPI$  mediante la variable  $TMedT_{obj}I$ . Estos parámetros se representan en las dos gráficas superiores de la Figura 7.1.2. También se registra el número de veces que se ha superado el tiempo objetivo, el cual se muestra en la gráfica inferior derecha de la Figura 7.1.2.

Si ordenamos el rendimiento de los algoritmos en orden descendente, según los AGV necesarios mediante  $NAGVD$ , el orden resultante será el siguiente:

1. El  $STPRAN$  y  $STPFIFO$ , en este entorno, aunque son los que tienen peor rendimiento, este no difiere mucho del resto de algoritmos. Esto lo vemos reflejado en dos factores: el primero es que en muchas rutas solo existe una única ruta posible; el segundo factor radica en que, aunque su gestión de intersecciones es reducida, al ser cruces sencillos es una gestión simple que no implica una gran diferencia de resultados, siendo capaces de realizar el trabajo con 28 AGV.
2. El  $STPACC$ , como en el Entorno 1, es capaz de mejorar el rendimiento de los algoritmos semejantes a él, gracias a una gestión ligeramente mejor en las intersecciones. Este algoritmo logra reducir el uso de los AGV en una unidad. Necesita 27 AGV.

3. El *ASTAR*, en este entorno, se ve ligeramente perjudicado, al calcular la ruta mediante  $A^*$ . En algunas rutas, este tipo de algoritmo genera una ruta ligeramente más larga y, aunque la diferencia es reducida, se ve aumentada porque genera mayor cantidad de tráfico. Necesita 26 AGV.
3. El *GEOPAR* al aplicar una arquitectura jerárquica presenta dificultades en este entorno, debido a su distribución, ya que las zonas no se pueden separar con facilidad, por lo que se genera una única zona en todo el mapa utilizando solo la mitad de algoritmo. Aun así, su rendimiento implica que se necesiten 26 AGV.
4. El *STPCT* tiene un desempeño elevado en este entorno respecto a los algoritmos anteriores e implica que se necesiten 25 AGV. Este algoritmo se ve favorecido en este entorno por dos motivos; el primero de ellos es que no se ve perjudicado por las rutas fijas, y el segundo es que, al tener un gestor de cruces más completo, puede unificar varios cruces sencillos en uno complejo, consiguiendo así una mayor tasa de paso de los AGV por las intersecciones.
5. El *TIMEWINDOW* es el algoritmo con mayor rendimiento y necesita solamente 24 AGV. Este mayor rendimiento está causado por dos factores, el primero, y principal, es la buena gestión de las intersecciones. El segundo factor, que es el que marca la diferencia, se encuentra en el cálculo de rutas alternativas teniendo en cuenta el tráfico. Aunque, en este entorno, en muy contadas ocasiones es significativo, es capaz de marcar la diferencia.

Si nos centramos en el siguiente parámetro de evaluación: el  $TMedT_{obj}$ , el orden de rendimiento se mantiene respecto a la anterior variable de cuantificación, y también podemos observar que, cuando un algoritmo dispone de un  $NAGVD$  cerca de su  $NAGV$  mínimo, el  $TMedT_{obj}$  sube y va acercándose a 0. Esto significa que los algoritmos, en un tipo de entorno con punto de salida único, son capaces de maximizar el rendimiento porque el retraso de un vehículo en el punto de inicio se suple con otro AGV. Esto nos lleva a deducir que, la utilización de un punto de salida único en un entorno que lo permita puede, por sí misma, reducir el número de AGV necesarios.

### 7.1.3 Entorno 3

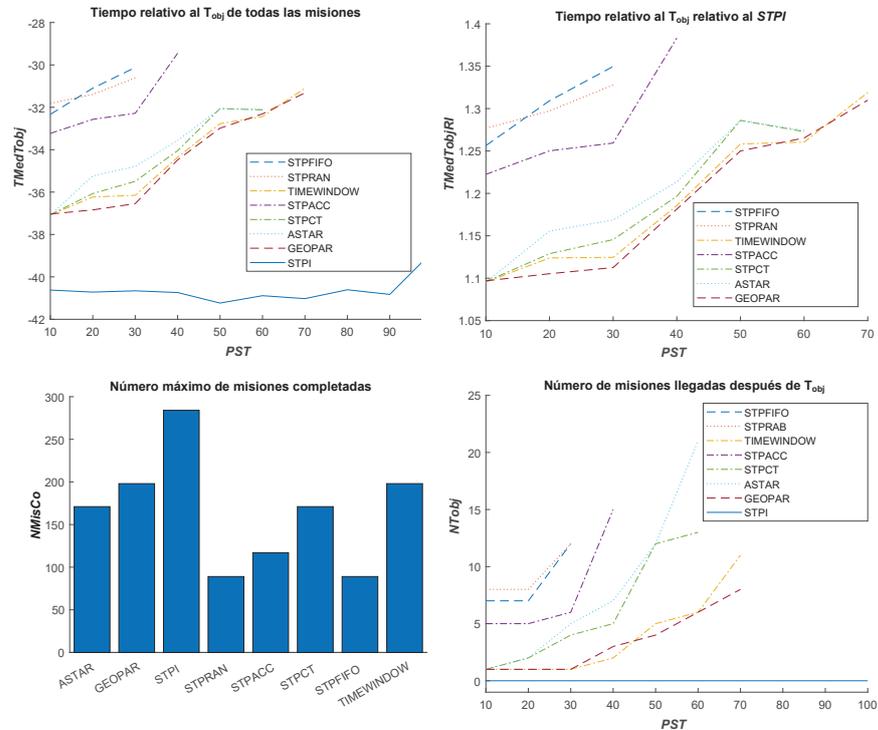


Figura 7.1.3: Resultado de los algoritmos de referencia del entorno 3

El último entorno de evaluación, expuesto en la Sección 6.4, es el más complejo de todos. Este entorno cuenta con diferentes islas, rodeadas de pasillos bidireccionales, que generan la posibilidad de abrir múltiples rutas para llegar a los destinos. Es por ello que, dependiendo del tráfico puntual, la mejor ruta que tomar variará a lo largo del tiempo. También es importante la gestión de las intersecciones, ya que se genera mucho tráfico en las intersecciones de dos pasillos bidireccionales. Es por ello que este entorno tiene dos factores muy relevantes para poder extraer un alto rendimiento: el cálculo de rutas teniendo en cuenta el tráfico y la gestión de las intersecciones. Utilizaremos el criterio de evaluación del entorno 3 y expondremos los resultados de estos criterios en la Figura 7.1.3. Los parámetros expuestos en las gráficas de la figura son los siguientes:

- **Gráfica superior izquierda:**
  - **Eje X:**  $PST$  (Porcentaje de carga de trabajo del  $Te$  respecto a tiempo de desplazamiento  $TTT_{areTask}$ ).
  - **Eje Y:**  $T_{MedT_{obj}}$  (Tiempo medio de llegada de los AGV a sus desti-

nos respecto a sus  $T_{obj}$  respectivos. Medido en segundos).

– **Valoración del eje Y:** Valor más bajo mejor.

• **Gráfica superior derecha:**

– **Eje X:**  $PST$  (Porcentaje de carga de trabajo del  $Te$  respecto a tiempo de desplazamiento  $TTTare_{Task}$ ).

– **Eje Y:**  $TMedT_{obj}RI$  (Valor de  $TMedT_{obj}$  respecto al obtenido por el algoritmo  $STPI$ . Medido en valor porcentual con respecto al algoritmo  $STPI$ ).

– **Valoración del eje Y:** Valor más bajo mejor.

• **Gráfica inferior izquierda:**

– **Eje X:** Algoritmos de referencia.

– **Eje Y:**  $NMisCo$  (Número de misiones completadas en un entorno, cumpliendo los requisitos del mismo, en un periodo de tiempo específico).

– **Valoración del eje Y:** Valor más alto mejor.

• **Gráfica inferior derecha:**

– **Eje X:**  $PST$  (Porcentaje de carga de trabajo del  $Te$  respecto a tiempo de desplazamiento  $TTTare_{Task}$ ).

– **Eje Y:**  $NT_{obj}$  (Número de veces que los AGV han llegado a sus destinos después del  $T_{obj}$ ).

– **Valoración del eje Y:** Valor más bajo mejor.

El valor más relevante para este entorno es el trabajo máximo que pueden realizar un número específico de AGV. Para obtener este valor, extraeremos el valor de  $NMisCo$  para cada  $PST$ . El valor que utilizaremos será el máximo que haya cumplido todos los requisitos, incluyendo el de que haya completado todas las misiones y que los AGV hayan llegado a todos sus destinos antes del  $Tmax$ .

Si ordenamos el rendimiento de manera creciente según el número máximo de misiones completadas, el orden será el siguiente:

1. El rendimiento de  $STPRAN$  y  $STPFIFO$  en este entorno es bajo, igual que ocurre en el resto de entornos. En este caso, salen perjudicados con el cálculo de rutas fijas y por su bajo rendimiento en la gestión de intersecciones. Estos algoritmos solo son capaces de lograr un  $PST$  del 30 %.
2. El  $STPACC$  tiene un rendimiento bajo, como los algoritmos parecidos a él. Solo conseguiremos un rendimiento ligeramente superior al lograr un  $PST$  del 40 % gracias a su mejor rendimiento en la gestión de intersecciones.
3. El  $ASTAR$ , en este entorno, al no tener en cuenta el tráfico en el cálculo de la ruta, ve reducido su rendimiento. Es capaz de un  $PST$  del 60 %.
3. Al  $STPCT$ , le ocurre lo mismo que al  $ASTAR$ . Al calcular rutas fijas, gen-

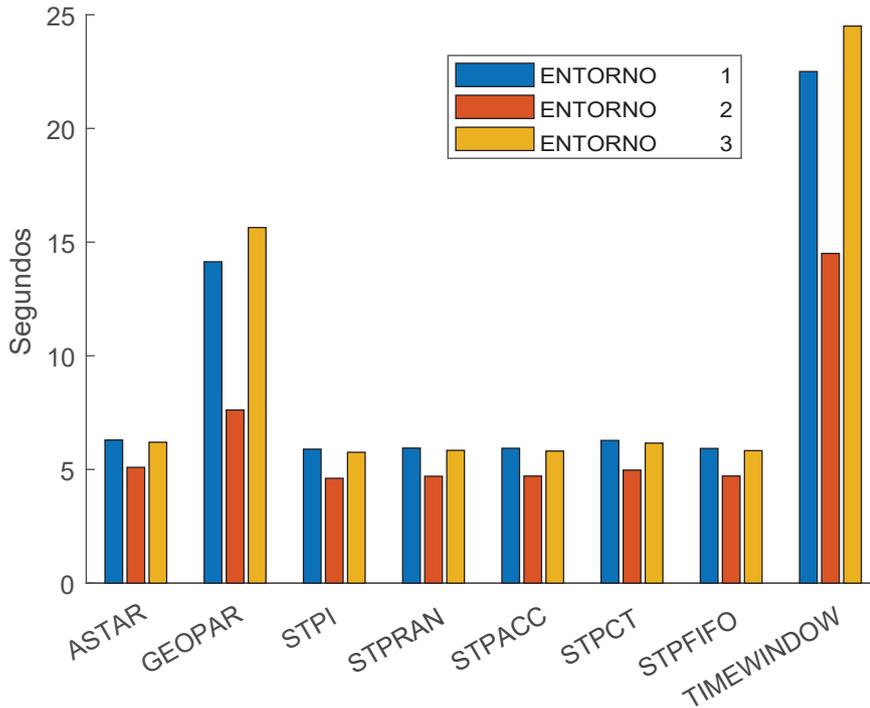
era mucho tráfico en algunas zonas. Solo varía ligeramente su rendimiento respecto a *ASTAR*, porque en este entorno su metodología de gestión de las intersecciones logra, en algunas ocasiones, un mejor cálculo. Esto implica un ligero mejor rendimiento en los criterios secundarios.

4. En este entorno, podemos aplicar la arquitectura jerárquica del *GEOPAR*, gracias a la cual será capaz de desarrollar todo su potencial al aproximarse al *Top Layer* del entorno real. El algoritmo *GEOPAR* es capaz de alcanzar un *PST* del 70 %.
4. En este entorno, el *TIMEWINDOW* es capaz de utilizar su metodología de cálculo de rutas teniendo en cuenta el tráfico, motivo por el cual es capaz de gestionar el tráfico eficientemente y logra un rendimiento de 70 % respecto al *PST*.

Si utilizamos las otras variables de evaluación: el  $TMedT_{obj}$ , el  $TMedT_{obj}RI$  y el  $NT_{obj}$ , podemos concretar aún más el rendimiento de los algoritmos. Si ordenamos el rendimiento respecto a  $NT_{obj}$ , el orden resultante será el siguiente:

1. Los algoritmos *STPRAN* y *STPFIFO* son los que tienen peor rendimiento en el número máximo de misiones completadas y, como es lógico, también son los que tienen un peor rendimiento bajo estos criterios de evaluación.
2. El *STPACC* se sitúa, con estos criterios de evaluación, entre los dos algoritmos anteriores y el resto de los algoritmos.
3. Aunque el algoritmo *ASTAR* tiene un número máximo de misiones que puede completar, igual que el *STPCT*, observamos que el  $TMedT_{obj}$  es igual o superior, para los diferentes valores de *PST*, que el *STPCT*; igual que ocurre con el  $NT_{obj}$ , denotando un rendimiento inferior respecto al *STPCT*.
4. Gracias a estas dos variables, observamos que la generación de rutas del *STPCT* es la causante de que este algoritmo sea capaz de obtener un rendimiento ligeramente superior que el del algoritmo *ASTAR*.
5. Si analizamos estas dos variables, observamos que el *TIMEWINDOW*, aun teniendo un desempeño muy elevado, en este entorno se ve superado ligeramente por el *GEOPAR*. Esto es debido principalmente a que, aunque el *TIMEWINDOW* tiene en cuenta el tráfico a la hora de calcular las rutas, su aproximación al cálculo de la posición futura de este algoritmo tiene una desviación considerable respecto a la realidad, cuando el tráfico es elevado.
6. El *GEOPAR*, en este tipo de entornos, puede explotar su potencial. Aquí podemos distinguir claramente las diferentes zonas. Este es el algoritmo con un mayor desempeño, al ser el que tiene mejores valores de  $TMedT_{obj}$  y  $NT_{obj}$ .

## 7.2 Carga computacional de los algoritmos de Gestión de flotas en cada entorno



**Figura 7.2.1:** Carga computacional de los 3 entornos, medido por el tiempo de cálculo necesario por hora (Valor más bajo mejor)

Un factor que valorar de un algoritmo es la carga computacional que se necesita para ejecutarlo, indicando el *hardware* necesario para ejecutar el algoritmo. Cuanto más carga computacional, más potente tendrá que ser el *hardware* donde se ejecute el algoritmo, incrementado el coste de su implantación.

Con el objetivo de comparar el cálculo computacional, calcularemos el tiempo requerido para extraer los resultados utilizando cada uno de los algoritmos. Aunque este valor no es directamente la carga computacional de los algoritmos, ya que incluimos la carga computacional necesaria para la extracción de los resultados, esta sí que se puede utilizar para evaluar las diferencias entre ellos, ya que todas han sido procesadas con el mismo *hardware*.

Si ordenamos el valor de la carga computacional expuesto en la Figura 7.2.1 para cada uno de los algoritmos en los 3 entornos, el orden ascendente de carga computacional será el siguiente:

1. El algoritmo con menos carga computacional es el *STPI*, que únicamente tiene que calcular la ruta más corta sin tráfico. La mayor parte del tiempo requerido para el cálculo ha sido utilizado para la extracción de resultados.
2. Los algoritmos más sencillos, como son el *STPRAN*, el *STPFIFO* y el *STPACC*, tienen una carga computacional muy baja, siendo esta ligeramente superior a la *STPI*.
3. Un nivel por encima se encuentran el *ASTAR* y el *STPCT*. Si los comparamos con el *STPI*, estos entornos ven su carga computacional aumentada entre un 5 % y un 10 %.
4. La carga computacional aumenta sustancialmente con el resto de los algoritmos, siendo el penúltimo algoritmo el *GEOPAR*, ya que al tener multitud de elementos que se encargan de gestionar las nuevas rutas y el tráfico de las diferentes partes, cada uno tomando sus decisiones, aumenta la carga computacional, sobre todo a la hora de generar rutas. Este algoritmo, respecto al *STPI*, tiene una carga computacional superior, entre un 150 % y un 300 % superior en estos entornos.
5. Finalmente, el Gestor de flotas que tiene una mayor carga computacional es el *TIMEWINDOW*. Esto es debido al cálculo constante de la posición futura de los vehículos sin contar con el tráfico, siendo esta utilizada para calcular la mejor ruta para cada AGV. La carga computacional se ve aumentada entre un 300 % y un 450 % respecto a *STPI*, en estos entornos.

Este valor es relevante por el hecho de que, a más carga computacional se requiere un *hardware* más potente, lo que supone un sobrecoste asociado. Este valor solo es relevante cuando tiene un rendimiento similar entre los distintos algoritmos, ya que el ahorro generado por el algoritmo suele ser mayor que el coste asociado a la carga computacional. Por otro lado, sí que es importante que la carga computacional tenga un ratio inferior a 1:1 con respecto a la realidad. Si no lo cumple, significa que no puede funcionar a tiempo real. En este caso, todos los algoritmos cumplen esta restricción.

### 7.3 Análisis de los algoritmos de Gestión de flotas

En este apartado, procederemos a analizar el rendimiento de los algoritmos mediante los resultados obtenidos. Con ellos, realizaremos una referencia entre los diferentes algoritmos, con sus ventajas y desventajas. Finalizaremos con las mejores características de cada uno de los algoritmos.

### 7.3.1 *STPRAN y STPFIFO*

Los algoritmos *STPRAN* y *STPFIFO* son ampliamente utilizados en la industria debido a su implementación más sencilla en comparación con otros algoritmos mencionados. Esto facilita tanto la implementación manual como automática en la generación de rutas.

Estos dos algoritmos tienen un comportamiento muy similar. En términos de rendimiento, aunque son inferiores a otros algoritmos, pueden manejar los vehículos de manera eficiente en entornos con una carga de trabajo media-baja. Son especialmente adecuados para entornos con poco tráfico debido a su facilidad de integración y su capacidad para operar de manera similar a otros algoritmos en estas condiciones.

### 7.3.2 *STPACC*

Este algoritmo es muy parecido a los dos algoritmos anteriores, ya que comparte con ellos el algoritmo de rutas. Es por ello, que tiene las mismas ventajas y desventajas que los anteriores en los diferentes entornos, obteniendo un valor superior en todos ellos gracias a su gestión de los cruces.

Debido a lo expuesto en el párrafo anterior, es más recomendable la utilización de este tipo de algoritmo que los dos anteriores, ya que su implementación es similar. La única consideración que se debe tener en cuenta es que hay que limitar el número de vehículos que pueden pasar continuamente, ya que, si no, se puede causar una reducción significativa del rendimiento con el incremento del tráfico.

### 7.3.3 *STPCT*

Este algoritmo es el máximo exponente de los algoritmos convencionales. Su principal característica es que realiza el cálculo de las rutas sin tener en cuenta el tráfico, motivo por el cual se ve perjudicado en mapas con muchas alternativas viables, ya que llevaría muchos vehículos por las mismas vías sin explotar las vías auxiliares. Pero, por otro lado, al contar con una gestión de cruces donde se tienen en cuenta todas las interacciones entre las diferentes entradas y conexiones internas de los cruces, es una característica que genera un rendimiento aceptable cuando el número de rutas es reducido.

Si analizamos el rendimiento en el primer entorno, que no se ve muy perjudicado al contar con rutas fijas, obtiene el máximo rendimiento, lo que lo equipara a algoritmos muchos más complejos. En el segundo entorno, donde lo más relevante es la gestión de los cruces, al unificar cruces simples creando complejos con el objetivo de tomar decisiones con un mayor número de factores

en juego, obtiene un muy buen rendimiento que solo se ve mejorado por un algoritmo. Finalmente, en el tercer entorno, que es mucho más complejo, es donde se aprecia la carencia en el cálculo de las rutas del algoritmo sin consideración del tráfico, obteniendo un rendimiento aceptable pero generando mucho tráfico en algunas zonas.

Observamos que la utilización del máximo exponente de un algoritmo convencional obtiene un buen rendimiento en entornos con complejidad media y baja, por lo que su utilización es muy aconsejable en este tipo de entornos, sin importar la carga de trabajo. Sus carencias solo se aprecian en entornos de complejidad alta, donde obtiene un resultado aceptable por su esfuerzo de integración.

#### 7.3.4 *AStar*

La principal ventaja de este algoritmo consiste en que está basado en  $A^*$ , lo que genera que el tiempo computacional en el cálculo de las rutas sea más rápido, sobre todo en mapas grandes. En el cómputo global obtiene un mejor rendimiento que los algoritmos convencionales, siendo su rendimiento similar al de *STPCT*, lo cual puede resultar extraño, ya que ambos están basados en diferentes algoritmos de cálculo de rutas:  $A^*$  y *Dijkstra*, respectivamente.

Este algoritmo se ve perjudicado en entornos con un *Roadmap* con muchas curvas y rutas unidireccionales, donde el cálculo mediante  $A^*$  lo lleva por rutas no óptimas.

Por los resultados obtenidos en estos entornos, al tener una carga computacional similar al de *STPCT*, es aconsejable la utilización de este otro algoritmo, ya que el rendimiento es, de media, ligeramente superior para el caso del *STPCT*.

#### 7.3.5 *GEOPAR*

Mediante los resultados de este algoritmo jerárquico, observamos que, de todos los algoritmos analizados, este es el que ve su rendimiento alterado en mayor medida en relación con el entorno, debido a la utilización del *Top Layer*. Cuando la aproximación del *Top Layer* es cercana a la del entorno, y existen muchas rutas posibles, en este entorno puede obtener el mayor rendimiento. Pero, por otro lado, cuando no es así, su rendimiento puede verse reducido, pudiendo obtener un rendimiento similar o inferior a los convencionales.

En estos casos, cuando el entorno lo favorece, la aproximación de enviar al vehículo a través de los *Top Layer*, dependiendo del tráfico de cada zona, puede

reportar un buen rendimiento cuando hay un tráfico denso.

El rendimiento del algoritmo depende en gran medida del entorno y su distribución, por lo que puede ser un algoritmo ideal en unos entornos y reportar poco rendimiento en otros.

### **7.3.6** *TIMÉWINDOW*

Este último algoritmo de referencia ha reportado, de media, un mejor rendimiento que los demás. Esto es debido a que su aproximación en el cálculo de las rutas, que tiene en cuenta el tráfico, se ve menos alterada por la morfología del entorno y del tráfico.

Este algoritmo es aconsejable en una gran variedad de entornos, viéndose solo perjudicado por su carga computacional y en entornos con rutas muy lejanas y con mucho tráfico.

A pesar de sus ventajas, este algoritmo presenta tres desventajas, las cuales, si se mejoran, podrían conducir a un rendimiento aún mayor:

- Cuando existe un tráfico elevado, su estimación de tráfico futuro se ve mermada porque la estimación de la posición futura del tráfico se realiza mediante cálculos que no tienen en cuenta el tráfico.
- El cálculo de la ruta se realiza al iniciar la misma, lo que genera que, cuanto más larga sea, peor será su gestión de las rutas, teniendo en cuenta el tráfico.
- Su carga computacional es elevada, al tener que ir calculando continuamente la posición futura de los vehículos.

## 7.4 Comparativa del rendimiento de los diferentes algoritmos de referencia

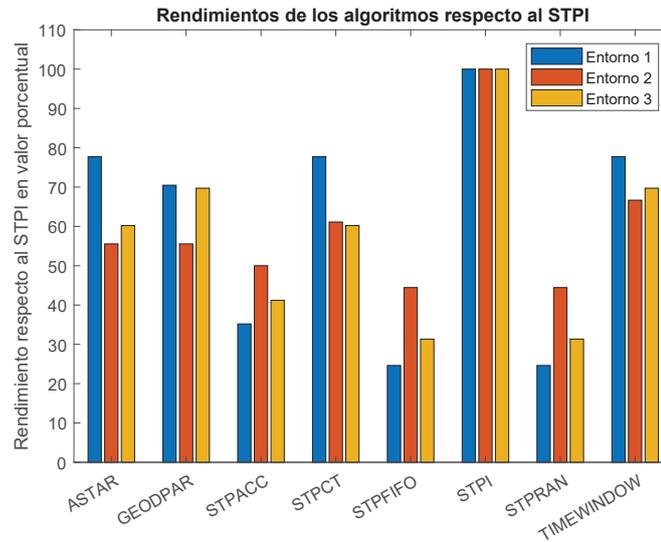


Figura 7.4.1: Resultado de los algoritmos de referencia del entorno 3

En este apartado visualizaremos las diferencias de rendimiento de los algoritmos de referencia. Para facilitar la lectura, por cada entorno, se ha calculado el rendimiento de cada algoritmo respecto al algoritmo de referencia de rendimiento máximo, el *STPI*, permitiendo visualizar con claridad el rendimiento. Mediante la Figura 7.4.1, se muestra el rendimiento de cada algoritmo respecto al rendimiento ideal en cada entorno.

Para calcular el rendimiento de los algoritmos de Gestión de flotas, se utiliza el indicador más relevante de cada entorno, y se compara respecto al obtenido por el algoritmo de referencia de rendimiento máximo, el *STPI*. Los indicadores utilizados para cada entorno son los siguientes:

- Entorno 1: Número máximo de misiones completadas, mediante la variable  $N_{MisCo}$ .
- Entorno 2: Número mínimo de AGV necesario para completar las misiones  $N_{AGV}$ .
- Entorno 3: Número máximo de misiones completadas, mediante la variable  $N_{MisCo}$ .

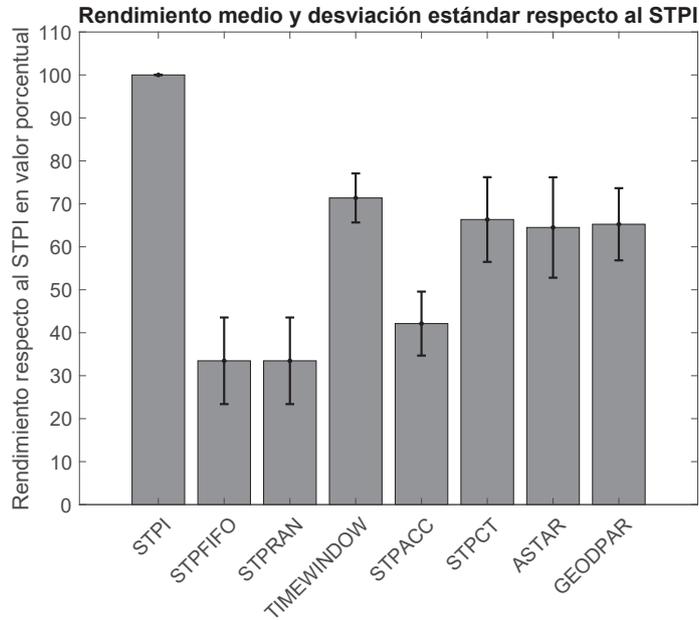


Figura 7.4.2: Resultado de los algoritmos de referencia del entorno 3

Para extraer un único indicador de rendimiento de cada algoritmo, en la Figura 7.4.2 y la Tabla 7.4.1 se indican el rendimiento medio de los 3 entornos respecto al rendimiento máximo ideal y su desviación estándar. Gracias a estos datos, podemos clasificar los algoritmos por su rendimiento.

Algoritmo	Media	Desviación Estándar
<i>STPFIFO</i>	33.47	10.08
<i>STPRAN</i>	33.47	10.08
<i>STPACC</i>	42.13	7.44
<i>ASTAR</i>	64.50	11.69
<i>GEOPAR</i>	65.25	8.40
<i>STPCT</i>	66.35	9.86
<i>TIMEWINDOW</i>	71.37	5.71
<i>STPI</i>	100.00	0.00

Table 7.4.1: Resultados de los algoritmos de comparación en los 3 entornos

Por lo tanto, si analizamos el rendimiento de cada uno de los algoritmos

utilizando este único valor, podemos ordenarlos según su rendimiento de la siguiente manera:

1. *STPRAN* y *STPFIFO*: Estos algoritmos, de fácil implementación, son los más comúnmente utilizados en la industria. Cuentan con la característica de que se pueden aplicar, indiferentemente, mediante un control centralizado o distribuido. Su escalabilidad es sencilla y además permiten la implementación manual de las rutas, pero su rendimiento se ve reducido en gran medida en entornos con tráfico medio y elevado.
2. *STPACC*: Este algoritmo es muy similar a los dos anteriores, con sus ventajas y desventajas, pero cuenta con un desempeño ligeramente mayor al manejar de forma más eficiente el tráfico.
3. *ASTAR*: Ha tenido un rendimiento aceptable en los diferentes entornos, ligeramente inferior a los otros algoritmos con mejor rendimiento. Con la desventaja de que, en entornos irregulares, su cálculo de rutas puede no ser tan óptimo.
4. *GEOPAR*: Es un algoritmo que puede reportar un gran rendimiento en entornos donde la aproximación de su *Top Layer* no difiera en exceso de la realidad. Tiene un rendimiento muy elevado en este tipo de mapas que, a su vez, cuentan con múltiples rutas posibles. Si el entorno no lo favorece, su rendimiento se ve en gran medida reducido. Su carga computacional y su dificultad de implementación son elevadas, siendo ideal para ser utilizado en entornos muy grandes y con múltiples rutas.
5. *STPCT*: Este es el máximo exponente de un algoritmo con metodologías convencionales. Gracias a la depuración de este algoritmo en entorno reales, aun al contar con algoritmos no muy complejos, logra extraer un buen rendimiento, siendo capaz de superar a otros algoritmos no convencionales. Obteniendo el segundo mejor rendimiento de esta comparativa.
6. *TIMEWINDOW*: Ha tenido buen rendimiento en todos los entornos analizados, logrando el mejor rendimiento entre todos los algoritmos de referencia, viéndose penalizado cuando las distancias de las rutas son elevadas. Su carga computacional es muy alta, al igual que su implementación.

Finalmente, si para finalizar la comparativa, en la siguiente Figura 7.4.3 se exponen de forma visual, las ventajas y desventajas:

Algoritmos	Ventajas	Desventajas	Rendimiento
<i>STPRAN</i> <i>STPFIFO</i> <i>STPACC</i>	Cálculo computacional bajo. Rápida implementación.	Rutas sin consideración del tráfico.	Rendimiento aceptable con poco tráfico. Bajo rendimiento con un tráfico medio y elevado.
<i>ASTAR</i>	Cálculo computacional bajo.	Problemas de cálculo de rutas en entornos con elementos divisorios grandes. Rutas sin consideración del tráfico.	Buen rendimiento en entornos con pocas rutas posibles. Rendimiento reducido en un entorno con múltiples rutas posibles.
<i>GEOPAR</i>	Gestión jerárquica, recalcula rutas en cada zona. Buena gestión de tráfico por zonas.	Cálculo computacional elevado. Problemas en entornos irregulares y con pocas rutas posibles.	Buen rendimiento en entornos regulares. Rendimiento reducido en entornos irregulares.
<i>STPCT</i>	Cálculo computacional bajo. Rápida implementación. Buena gestión de las intersecciones.	Rutas sin consideración del tráfico.	Buen rendimiento en entornos con pocas rutas posibles. Rendimiento reducido en un entorno con múltiples rutas posibles.
<i>TIMEWINDOW</i>	Rutas con consideración del tráfico. Estimación del tráfico con ventanas temporales.	Cálculo computacional elevado. No recalcula las rutas durante el trayecto.	Buen rendimiento en multitud de entornos diferentes. Rendimiento reducido en un entorno con rutas largas.

Figura 7.4.3: Ventajas y desventajas de los algoritmos de comparación

## 7.5 La mejor combinación

De este análisis se pueden extraer diferentes características de los algoritmos de Gestión de flotas de referencia, que han demostrado buenos resultados. Principalmente, se han identificado dos campos de actuación donde existe potencial de mejora al adaptar los algoritmos al entorno:

- **Gestión del tráfico en las intersecciones:** La modelización de las intersecciones del algoritmo *STPCT* ha mostrado un buen rendimiento en este campo. La mejora a explotar en este campo radica en la selección óptima del AGV que espera en la intersección para determinar cuál debe pasar

en cada momento. Esta elección adecuada del vehículo puede generar una mejora significativa en la gestión del tráfico en las intersecciones y será una de las claves que se centrará esta tesis.

- **Generación de rutas teniendo en cuenta el tráfico:** En este campo, se han identificado los siguientes algoritmos que consideran el tráfico al generar las rutas:
  - *TIMEWINDOW*: Al estimar la posición futura de todos los AGV, este algoritmo puede utilizar dicha información para calcular rutas eficientes. Sin embargo, se ve penalizado en rutas muy largas y por su estimación de la posición futura de los AGV, que no considera el tráfico real. Si se aborda este factor, si se mejora como se estima la posición futura de los AGV podría lograrse una mejora en el rendimiento del algoritmo.
  - *GEOPAR*: Este algoritmo recalcula la ruta en cada zona y tiene en cuenta el tráfico mediante el uso del *top layer*. Sin embargo, su aproximación en el *top layer* al entorno no siempre es precisa debido a las peculiaridades de dicho entorno. Además, el cálculo en el *top layer* solo considera la ocupación y no el tráfico real generado. Estos son inconvenientes a tener en cuenta para mejorar el rendimiento del algoritmo.

De estas pruebas se extraen dos campos de actuación para realizar esta tesis. El primero de ellos es la incorporación de la gestión de las intersecciones basada en el algoritmo *STPCT*, pero adaptada al entorno y que seleccione el AGV teniendo en cuenta los indicadores específicos de cada entorno. El segundo campo es la generación de rutas basada en el algoritmo *TIMEWINDOW*, pero con adaptación al entorno y una estimación de la posición futura que tenga en cuenta el tráfico y las intersecciones.

## Capítulo 8

# Algoritmo generador de gestores de flotas: Fase I - Entorno simplificado

### Resumen

En este capítulo se busca demostrar la viabilidad de utilizar un algoritmo generador de metodologías que se adapte al entorno, teniendo en cuenta las peculiaridades del entorno y los criterios de evaluación. Por lo tanto, se presentará una primera aproximación de un algoritmo capaz de generar un Gestor de flotas adaptado a un entorno simplificado.

Si partimos de la base de que en este tipo de entornos se pretende incluir una gran cantidad de AGV, el factor más relevante para obtener un buen rendimiento de estos será la gestión adecuada del tráfico generado por los AGV y los elementos externos. El lugar donde se genera la mayor cantidad de tráfico son las intersecciones, por lo que este capítulo se centrará en mejorar el rendimiento en una intersección aislada.

Como primera aproximación en esta Fase I, se procederá a exponer un algoritmo que sea capaz de generar un Gestor de flotas adaptado al entorno simplificado, contando con un solo cruce con múltiples entradas y salidas. Llamaremos a este algoritmo *GA1*. Además, se comparará su rendimiento con otros

tipos de algoritmos de gestión de intersecciones.

Los apartados expuestos en este capítulo son los siguientes:

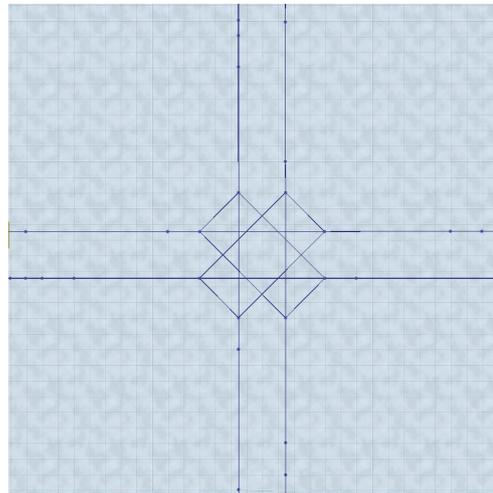
- **Definición del entorno:** Se definirá el entorno simplificado donde se procederá a evaluar y realizar la adaptación del Gestor de flotas generado.
- **Algoritmos de referencia:** Se expondrán los algoritmos de gestión de intersección de referencia para evaluar el rendimiento del algoritmo *GA1*.
- **Algoritmo *GA1*:** Se presentará el algoritmo *GA1*.
- **Resultados:** Se evaluarán los resultados obtenidos al generar un Gestor de flotas mediante el algoritmo *GA1* en el entorno simplificado y se comparará su rendimiento con los algoritmos de gestión de intersección de referencia.

## 8.1 Definición del entorno

Al ser un entorno sencillo, tendrá una única intersección. El primer paso será definir el tipo de intersección del entorno. Finalmente, concluiremos con los métodos de cuantificación utilizados.

### 8.1.1 Definición de la intersección

En el ámbito de la Gestión de flotas de los AGV, una intersección o cruce es un lugar donde hay, al menos, dos entradas posibles y una salida, en la que, si dos o más vehículos quieren entrar al cruce, es necesaria una política o algoritmo que determine el paso para evitar que los vehículos colisionen o se queden inmovilizados.



*Figura 8.1.1:* Intersección de dos corredores bidireccionales con caminos múltiples

Existen muchos tipos de intersecciones: con diferentes números de entradas y salidas de vehículos, con carriles unidireccionales o bidireccionales, etc. Es por ello que hay una gran variedad de elección entre los cruces a utilizar. Desde uno muy simple con dos entradas y una salida, hasta intersecciones complejas con muchas entradas y salidas posibles. Como en la industria es común el uso de islas, las intersecciones más comunes son de dos pasillos que interaccionan perpendicularmente, pudiendo ser unidireccionales o bidireccionales. En este caso, seleccionaremos el bidireccional, ya que el unidireccional es más sencillo y no genera tanto tráfico.

En este caso, la intersección propuesta (véase la Figura 8.1.1) cuenta con

cuatro entradas y cuatro salidas posibles, donde el destino de un vehículo entrante pueden ser todas las demás salidas, menos por la que está entrado.

### 8.1.2 Definición del entorno de evaluación

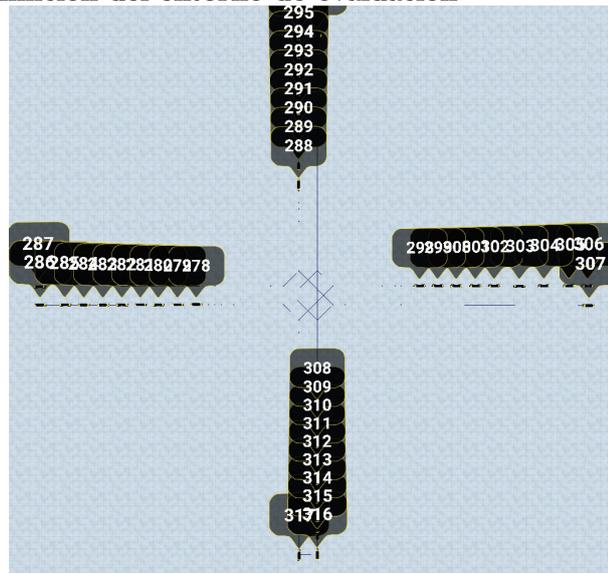


Figura 8.1.2: Estado inicial de la intersección del caso de uso de la FASE I

En el entorno de la intersección anteriormente expuesta, introduciremos 40 vehículos autónomos, como se observa en las Figuras 8.1.2 y 8.1.3. Estos vehículos pertenecen al modelo K11 de la empresa Kivnon, como se muestra en la Figura 6.4.4. Inicialmente, 10 de ellos estarán posicionados en cada una de las entradas de la intersección, tal como se muestra en la Figura 8.1.2.

Antes de llegar al inicio de la intersección, cada AGV ya tendrá asignado su destino objetivo. De esta forma, cada AGV sabrá de antemano a qué salida del cruce tiene que dirigirse, que puede ser cualquiera de las otras salidas. La asignación de la ruta se realiza de forma aleatoria. Una vez que el AGV sale del cruce, se dirige a la entrada correspondiente a esa rama del cruce.

Para cada simulación, definiremos un número determinado de AGV que pretenden entrar en el cruce por minuto. El conflicto generado por las peticiones entrantes se verá aumentado con el incremento de peticiones de entrada de los AGV. Lanzaremos al mismo tiempo las misiones de todas las entradas a los cruces, con el objetivo de generar una mayor dificultad en la gestión del tráfico.

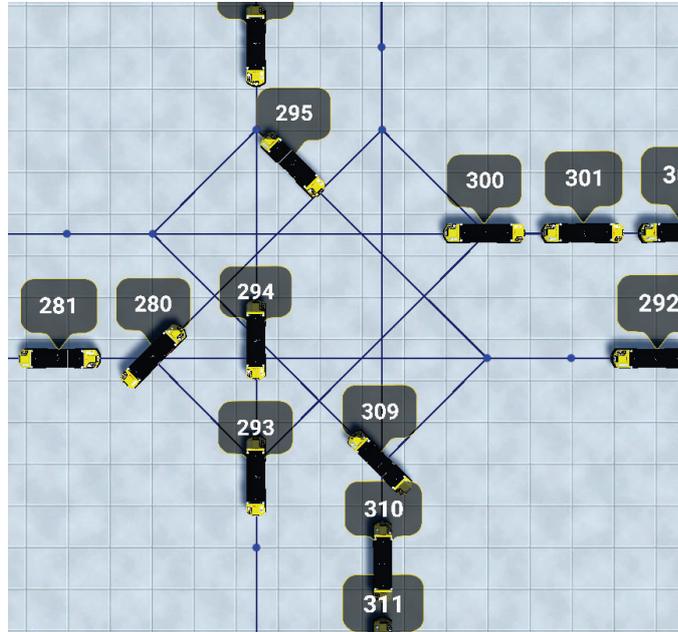


Figura 8.1.3: Estado inicial de la intersección del caso de uso de la FASE I en detalle

El número de peticiones de entrada de los AGV partirá desde un valor que genere un tráfico medio y lo iremos aumentando hasta que cada uno de los algoritmos logre su máximo número de vehículos que logran pasar por el cruce en un tiempo definido y fijando el tiempo de evaluación.

### 8.1.3 Cuantificación

Con el objetivo de evaluar las diferentes políticas o algoritmos que van a gestionar una intersección o cruce, necesitamos definir qué parámetros resultantes se indicarán en el rendimiento de los algoritmos. En este caso, solo estamos evaluando un cruce, por lo que un dato muy relevante consiste en el número total de AGV que ha logrado atravesar el cruce durante todo el tiempo de la evaluación, lo cual determina, directamente, el rendimiento del cruce.

## 8.2 Algoritmos de referencia

En esta fase inicial, no se propone un algoritmo completo de Gestión de flotas, sino que solo se pretende evaluar la gestión de este en una intersección, por lo que propondremos la gestión de las intersecciones de los siguientes algoritmos de referencia. El listado de estos algoritmos es el siguiente:

- *STPRAN*
- *STPFIFO*
- *STPACC*
- *STPCT*

### 8.3 Algoritmo GA1

En esta fase, al no proponer un algoritmo completo, y siendo el objetivo de esta prueba el aprendizaje de un cruce, expondremos un algoritmo que solo determine el paso de los AGV de un cruce. Pero como un Gestor de flotas realiza más tareas, en el apartado de cálculo de rutas estas estarán predefinidas, ya que, en este entorno, solo existe una ruta posible. En las siguientes fases profundizaremos más en la extracción de rutas. Nombraremos este algoritmo propuesto como *GA1*.

[Redacted text block]

### 8.3.1 Aprendizaje

[Redacted text block]

#### 8.3.1.1 Inicialización

[Redacted text block]

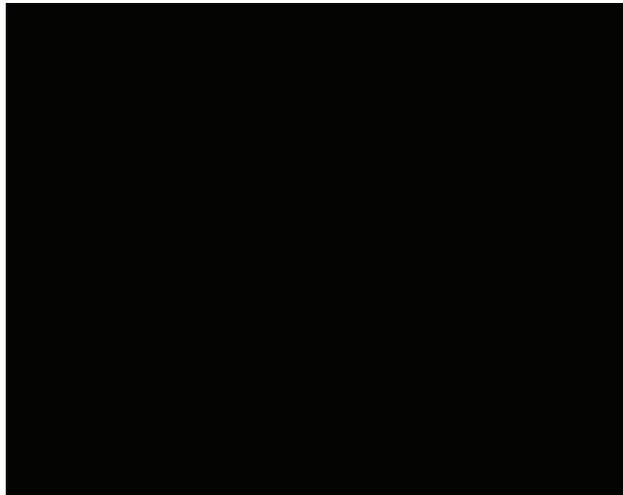
8.3.1.2 Evaluación

[Redacted text block]

### 8.3.1.3 Selección

[Redacted text block]

[Redacted text block]



[Redacted text block]

8.3.1.4 Recombinación

[Redacted text block]

### 8.3.1.5 Mutación

[Redacted text block]

8.3.1.6 Evolución

[Redacted text block]

### 8.3.1.7 Algoritmo de aprendizaje



*Figura 8.3.2:* Diagrama del aprendizaje de GA1



[Redacted text block]

**8.3.2 Modelización del entorno**

[Redacted text block]



*Figura 8.3.3:* Ejemplo de una tabla de segmentos

[Redacted text block]

**8.3.3 Extracción de los datos**

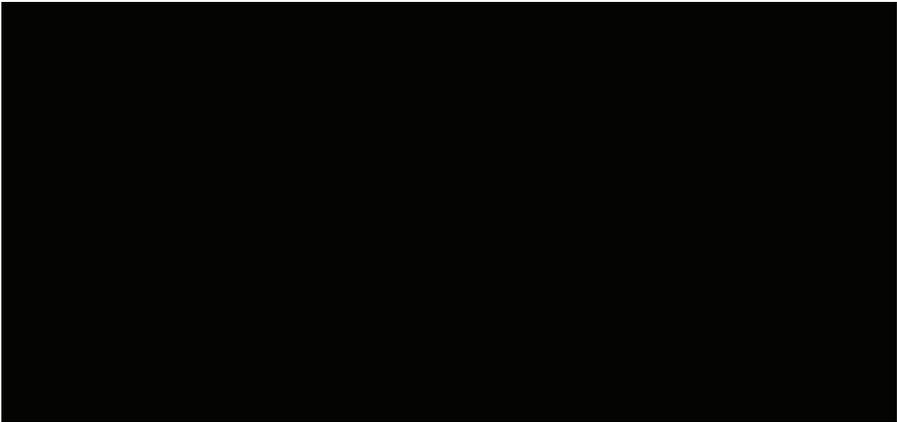
[Redacted text block]

**8.3.4 Fase de selección**

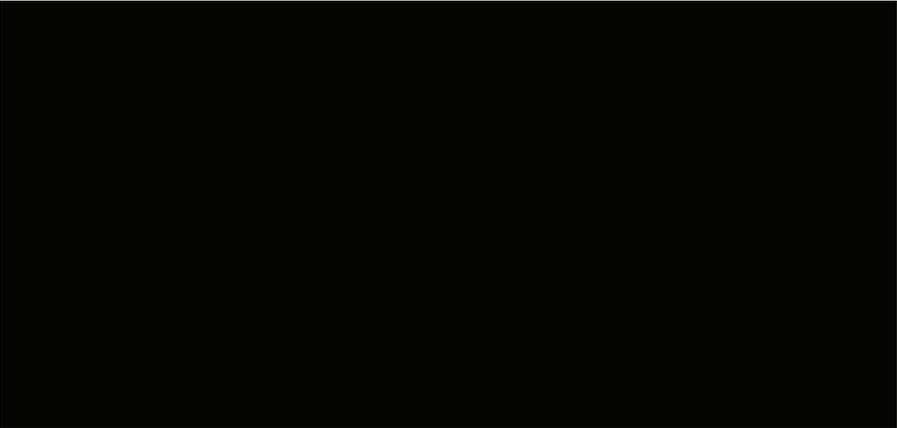
[Redacted text block]

### 8.3.5 Fase de comprobación

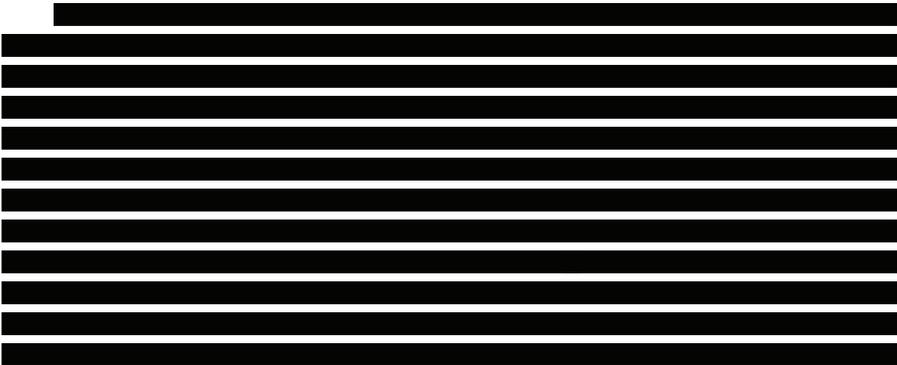
[Redacted text block]



*Figura 8.3.5:* Ejemplo de una tabla de colisiones de la intersección de la Figura 8.3.4



*Figura 8.3.4:* Ejemplo visual de la representación de una intersección



[Redacted text block]

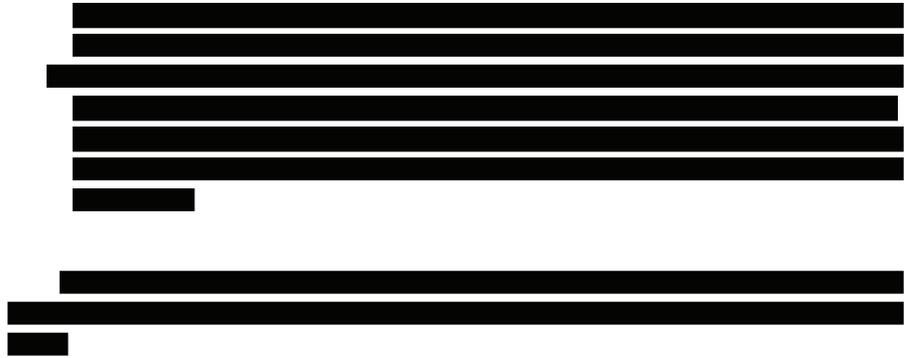
### 8.3.6 Algoritmo completo



*Figura 8.3.6:* Diagrama del algoritmo de la Fase I



[Redacted text block]



### 8.4 Resultados

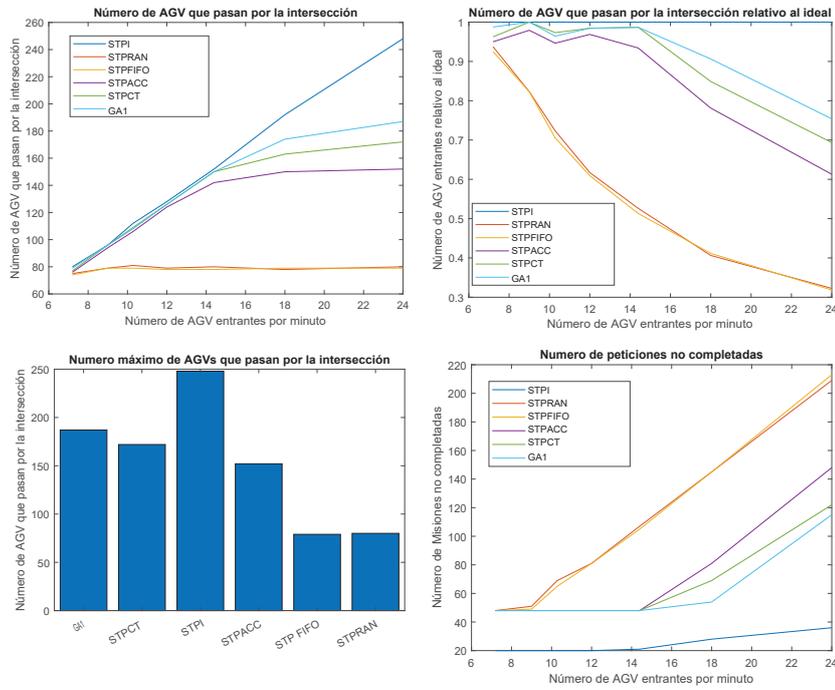


Figura 8.4.1: Resultados del número máximo de AGV que pasan por la intersección de la Fase I

Después de emular todas las políticas y algoritmos propuestos de gestión de intersecciones en las condiciones de simulación explicadas anteriormente, pro-

cederemos a la evaluación de los algoritmos de referencia, así como del algoritmo propuesto *GA1*, mediante la figura 8.4.1.

En la gráfica inferior izquierda de la figura 8.4.1, se indica el número máximo de vehículos que han logrado cruzar por la intersección durante el periodo de la simulación entre todas las peticiones de entradas de AGV por minuto. De aquí extraeremos un indicativo único de comparación entre algoritmos en el que, cuanto mayor sea el número de vehículos que han logrado cruzar, mejor rendimiento obtendrá el algoritmo. Por lo que, si ordenamos el rendimiento de los diferentes algoritmos de manera ascendente, el orden resultante será el siguiente:

- Las políticas de gestión de cruce de *STPI* y *STPRAN* son las que tienen peor rendimiento, esto es debido a que solo dejan cruzar un algoritmo a la vez.
- La siguiente política de gestión de cruce de *STPACC* consigue un mejor rendimiento, al permitir el paso seguido por la misma entrada. En este caso, la limitación del número máximo de AGV que pueden entrar continuamente por un nodo de entrada aumenta en gran medida el rendimiento de este algoritmo.
- El algoritmo de gestión de cruces *STPCT* ha obtenido un buen resultado, ya que deja pasar a los AGV que no colisionarían. Esta es la misma política que la de la fase de comprobación del *GA1*.
- El algoritmo genético propuesto *GA1* obtiene el mejor resultado, ya que se adapta al tipo de cruce mediante aprendizaje.

Otro parámetro que podemos utilizar para evaluar es el número de AGV capaces de cruzar dependiendo de las peticiones entrantes. Véase en la gráfica superior izquierda de la Figura 8.4.1 y en la gráfica superior derecha de la Figura 8.4.1 respecto al algoritmo sin consideración del tráfico. Gracias a este valor, podemos extraer más información del rendimiento de los algoritmos en diferentes situaciones. Si ordenamos el rendimiento de los algoritmos de manera creciente, el orden resultante será el siguiente:

- En las políticas de gestión de cruce de *STPI* y *STPRAN* observamos que el máximo de AGV que pasan a través de la intersección durante la simulación llega a su máximo de 80 cuando entran 10 AGV al cruce por minuto, no siendo capaz de gestionar el paso de más AGV.
- En la política de gestión de cruce de *STPACC*, al añadirle entradas acumuladas, obtiene un mayor rendimiento. Hasta 12 AGV por minuto de entrada obtiene un rendimiento del 95 % respecto al algoritmo sin consideración de tráfico.
- El algoritmo de gestión de cruces *STPCT* tiene un rendimiento superior

al 97 % hasta un número de 14 AGV entrantes. A diferencia de los otros algoritmos, llegados a este punto siguen aumentando el número de AGV entrantes con una pendiente inferior.

- El algoritmo genético propuesto *GA1* y el algoritmo *STPCT* tienen un rendimiento superior al 97 % hasta un número de 14 AGV entrantes. A diferencia de los otros algoritmos, llegados a este punto siguen aumentando el número de AGV entrantes con una pendiente inferior. Finalmente, obtiene un valor sustancialmente mejor al de *STPCT* con el aumento de tráfico.

### 8.4.1 Resultados del aprendizaje

Si analizamos el ritmo de aprendizaje del algoritmo propuesto para el caso con mayor número de vehículos entrantes, observamos que es necesaria una media de 50 generaciones para alcanzar el rendimiento máximo, como se puede observar en la Figura 8.4.2.

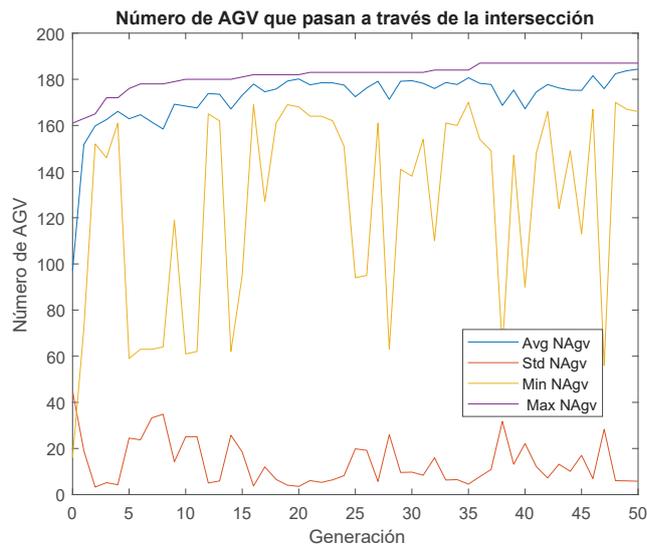


Figura 8.4.2: Número de vehículos que atraviesan una intersección durante el aprendizaje de *GA1* (Valor más bajo mejor)

### 8.4.2 Análisis de los resultados

Después de analizar los datos de los resultados, observamos que la generación de un gestor del tráfico adaptado a un entorno puede suponer una mejora significativa en la eficiencia. Por lo tanto, agregar esta metodología de generación de

gestor del tráfico adaptado a un entorno a un *Gestión de flotas* completo puede maximizar la eficiencia de la gestión de una intersección.

## Capítulo 9

# Algoritmo generador de gestores de flotas: Fase II - Entorno completo

### Resumen

En este capítulo se presenta el algoritmo *GA2*, el cual es capaz de generar un Gestor de flotas adaptado a un entorno completo.

Tras verificar que el algoritmo *GA1* presentado en la Sección 8 muestra un buen rendimiento debido a la adaptación de la gestión de las intersecciones al entorno, en esta Fase II se desarrollará un algoritmo que sea capaz de realizar una gestión de flotas adaptada a un entorno completo, centrándose en la adaptación de la gestión de las intersecciones al entorno.

Este entorno incluirá múltiples intersecciones, tipos de entrega, tareas de recogida y entrega en localizaciones con varias estaciones, y elementos externos a la simulación que pueden entorpecer a los AGV. Requiriendo que el Gestor de flotas generado por el *GA2* cumpla con todas las responsabilidades de la Gestión de flotas.

Para el algoritmo *GA2*, se utilizarán metodologías de algoritmos tradi-

cionales para todas las funcionalidades que no sean para la gestión de las intersecciones.

La exposición de la Fase II se divide en los siguientes apartados:

- **Definición del entorno:** Definición de los entornos utilizado para el aprendizaje y evaluación del algoritmo *GA2*.
- **Algoritmos de referencia:** Presentación de los algoritmos de gestión de flotas utilizados como referencia.
- **Algoritmo GA2:** Descripción del algoritmo propuesto en esta sección.
- **Resultados:** Resultados del rendimiento del algoritmo GA2 y su comparación con los algoritmos de referencia presentados.

## 9.1 Definición del entorno

Para este caso de uso, se llevará a cabo el aprendizaje y posterior evaluación de *GA2* en el entorno más completo y desafiante. Con este propósito, se utilizará el tercer entorno descrito en la Sección 6.4.

## 9.2 Algoritmos de referencia

En esta fase, propondremos un generador de Gestor de flotas que sea capaz de gestionar una flota de AGV en un entorno completo. A diferencia de la fase anterior, en este apartado, al contar con un entorno completo, podremos comparar el rendimiento con algoritmos completos. Los algoritmos de referencia para esta Fase II serán los expuestos en el Capítulo 5:

- *STPRAN*
- *STPFIFO*
- *STPACC*
- *STPCT*
- *GEOPAR*
- *ASTAR*
- *TIMEWINDOW*

## 9.3 Algoritmo GA2

Anteriormente, en la Fase I, se propuso el algoritmo *GA1*, que era capaz de generar un algoritmo de gestión para un solo cruce. En esta Fase II, proponemos el algoritmo *GA2*, que tiene como objetivo, basándose en el algoritmo *GA1*, generar un Gestor de flotas que sea capaz de realizar una gestión completa de vehículos en un entorno con múltiples cruces, peticiones y teniendo que generar la ruta de todos los vehículos.

Si nos centramos en la generación de rutas, para simplificar el desarrollo de este algoritmo, utilizaremos un algoritmo de cálculo de rutas que sea capaz de calcular la ruta óptima sin tráfico. Para ello, utilizaremos el cálculo de rutas propuesto en el algoritmo sin restricción de tráfico expuesto en la Sección 3.3.



[Redacted text block]

### 9.3.1 Metodología de aprendizaje

[Redacted text block]

#### 9.3.1.2 Evaluación

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

**9.3.1.3 Evolución**

[Redacted text block]

#### 9.3.1.4 Algoritmo de aprendizaje



*Figura 9.3.1:* Diagrama del aprendizaje de GA2



[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

**9.3.2 Extracción de los datos**

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

### 9.3.3 Algoritmo completo



*Figura 9.3.2:* Diagrama del algoritmo **GA2** de la Fase II



[Redacted text block]

## 9.4 Resultados

Para extraer el rendimiento de este algoritmo propuesto, lo compararemos con el resto de algoritmos de referencia en el tercer entorno propuesto, ya que es el más complejo para los Gestores de flotas. El criterio de evaluación será el expuesto en la sección 6.4 para este entorno 3. Solo expondremos el rendimiento del algoritmo propuesto *GA2* respecto al resto de algoritmos. El rendimiento del resto de algoritmos queda expuesto en el apartado de la Sección 7.1.3.

### 9.4.1 Comparativa con los algoritmos de referencia

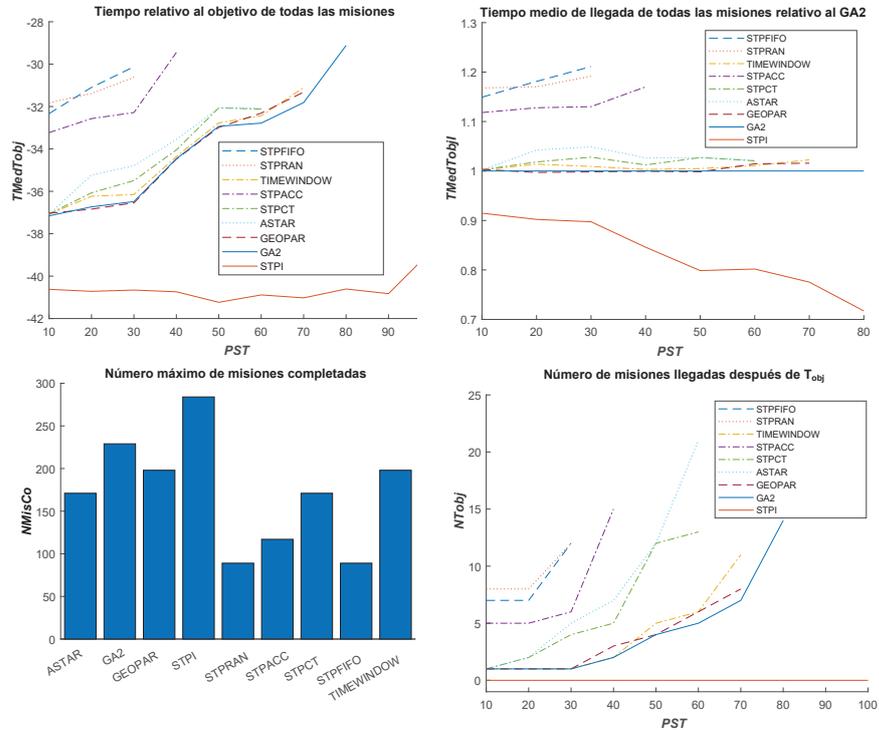


Figura 9.4.1: Resultados de la Fase II en el entorno 3

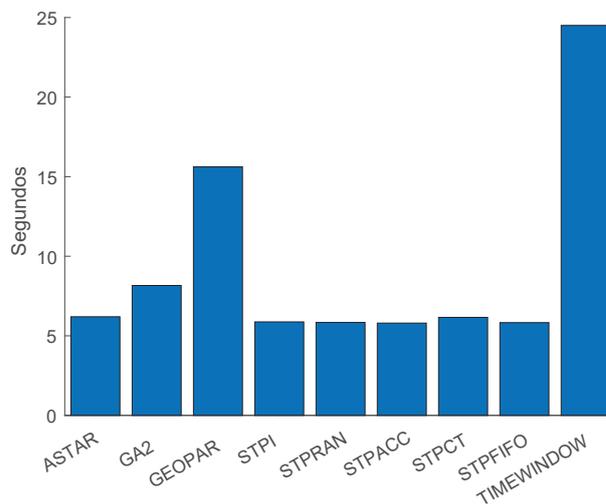
Una vez presentado el algoritmo *GA2*, procederemos a comparar su rendimiento en relación con los algoritmos de referencia del tercer entorno, descrito en la Sección 6.4, junto con los criterios de evaluación específicos para este entorno.

Si analizamos el resultado de las simulaciones de la gráfica inferior derecha de la Figura 9.4.1, podemos desgranar el rendimiento de los diferentes algoritmos dependiendo de la carga de tráfico de las simulaciones. En el caso del algoritmo propuesto en esta Fase II, con una carga del 30 % respecto al algoritmo sin consideración del tráfico, obtiene el mismo resultado que los mejores algoritmos en este apartado: el *GEOPAR* y el *TIMEWINDOW* y, en esta situación, solo un AGV llegó después del tiempo objetivo. Con una carga de tráfico medio de hasta el 50 %, tiene un rendimiento similar, o ligeramente superior, a estos algoritmos. Pero después del 50 % respecto al algoritmo sin consideración del tráfico, el rendimiento de este algoritmo propuesto empieza a diferenciarse respecto a estos dos algoritmos. El *GA2* es el único capaz de tener unos resultados satisfactorios al 80 %, ya que los otros dos algoritmos no han podido completar

todas las tareas satisfactoriamente, al llegar alguno de los vehículos más tarde del tiempo  $T_{max}$ .

Finalmente, en las dos gráficas superiores de la Figura 9.4.1, observamos el  $T_{MedT_{obj}}$  y el  $T_{MedT_{obj}RI}$ , si comparamos el tiempo medio del trayecto en relación con  $NT_{obj}$  (véase la gráfica inferior derecha analizada en el párrafo anterior), podemos analizar el rendimiento medio del trayecto respecto a su desviación. El rendimiento, en este apartado del algoritmo propuesto  $GA2$ , es muy similar al del  $GEOPAR$ , hasta un  $PST$  del 50 %. Después de este valor, el algoritmo propuesto obtiene un rendimiento mayor.

### 9.4.2 Carga computacional



**Figura 9.4.2:** Carga computacional de la Fase II en el entorno 3, medido por el tiempo de cálculo necesario por hora (Valor más bajo mejor)

Computacionalmente, este nuevo algoritmo tendría que ser similar al  $STPCT$ , ya que comparte con él la manera de encontrar la ruta óptima, así como parte de la gestión de las rutas, con la diferencia de que incorpora el aprendizaje. Este valor de carga computacional del  $GA2$  lo podemos observar en la Figura 9.4.2. Analizando el valor del algoritmo  $GA2$ , claramente observamos un incremento respecto al  $STPCT$ , debido a los cálculos extra necesarios en la toma de decisiones. La carga computacional de este algoritmo respecto al algoritmo sin consideración del tráfico es de alrededor del 40 % superior para este entorno. Un valor bastante bajo si lo comparamos con los algoritmos más complejos.

### 9.4.3 Aprendizaje

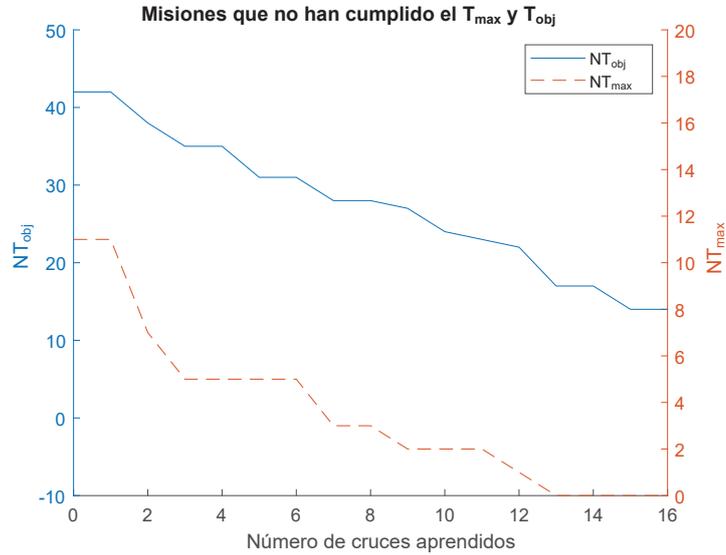


Figura 9.4.3: Resultados durante el aprendizaje de GA2 (Valor más bajo mejor)

El algoritmo propuesto GA2 realiza un aprendizaje de los cruces de manera secuencial, por lo que, si nos fijamos en la Figura 9.4.3, podemos ver el número de misiones que han llegado después del tiempo  $T_{obj}$  y  $T_{max}$ . En este caso, hemos extraído el aprendizaje con una carga de trabajo de  $Te$  del 80 % respecto al algoritmo sin consideración del tráfico.

Si observamos la Figura 9.4.3, el algoritmo GA2, al estar empezando a aprender y no tener ningún cruce aprendido, obtiene un rendimiento muy bajo, y al ir realizando el aprendizaje de las intersecciones, ve su número de  $NT_{obj}$  y  $NT_{max}$  reducido. Se aprecia que el incremento del rendimiento al realizar el aprendizaje de cada cruce es variable, debido a que las intersecciones que tienen mayor índice de tráfico tienen mayor potencial de mejora al tener mayor impacto en el rendimiento global.

El algoritmo propuesto es capaz de lograr satisfactoriamente un valor de  $PST$  mayor respecto a los demás algoritmos y, si analizamos la causa, esta no tiene que ver con el cálculo de las rutas de los AGV, ya que el cálculo de rutas es el utilizado en otros algoritmos con menor rendimiento. Por el contrario, el buen rendimiento es debido al aprendizaje de las intersecciones que, a pesar de realizar un aprendizaje secuencial, al utilizar valores totales de evaluación del

entorno, es capaz de minimizarlos.

#### 9.4.4 Análisis de los resultados

La conclusión que podemos extraer de estos resultados es, que la aplicación del aprendizaje en la gestión de las intersecciones puede reportar un buen rendimiento, sobre todo cuando el tráfico es elevado, porque canaliza las prioridades de los AGV según los criterios de evaluación de cada entorno. Estos resultados indican un gran potencial de mejora de rendimiento en la aplicación del aprendizaje de las intersecciones para la Gestión de flotas.

Existen potenciales vías de mejora para el algoritmo propuesto. Son las siguientes:

- Mejora de la gestión de las intersecciones del algoritmo *GA2*. Una posible mejora sería la inclusión de todos los datos de las demás intersecciones con el objetivo de poder tomar una decisión de tráfico teniendo en cuenta el estado de las mismas. Otra mejora consistiría en incluir en los datos de decisión variables de evaluación del entorno.
- Incorporar en el cálculo de rutas la afectación del tráfico es un aspecto que también consideran otros algoritmos de la comparativa. Esta mejora en el cálculo de rutas puede proporcionar un considerable incremento en el rendimiento.
- Mejora del proceso de aprendizaje. Otra posible mejora consiste en cambiar de un aprendizaje secuencial a uno global.



## Capítulo 10

# Algoritmo generador de gestores de flotas: Fase III - Algoritmo completo

### Resumen

En este capítulo se presenta el generador de Gestores de flotas adaptados al entorno, denominado *GA3*. A diferencia del algoritmo propuesto en la Fase II, *GA2*, que puede adaptarse solo a un entorno específico (el tercer entorno descrito en la Sección 6.4), en este caso se introduce un generador capaz de adaptarse a diferentes entornos, aunque se adapta a uno a la vez.

También se introduce una metodología de cálculo de rutas basada en el algoritmo *TIMEWINDOW* [13]. Este algoritmo calcula el trayecto añadiendo tiempo a las rutas, considerando la estimación de la circulación de otros AGV. La posición futura del vehículo se determina mediante un cálculo dinámico, sin tener en cuenta el posible tráfico generado. En el algoritmo *GA3*, se propone mejorar la estimación de la posición futura mediante la emulación del movimiento de los AGV, teniendo en cuenta la dinámica del vehículo, las detenciones causadas por los elementos de anticollisión (generalmente ocasionadas por el tráfico en el mismo sentido y ocasionalmente para evitar colisiones) y las detenciones causadas por el gestor de intersecciones (adaptado al entorno).

Al incluir el gestor de intersecciones adaptado al entorno en el cálculo de las rutas y llevar a cabo el aprendizaje del gestor de intersecciones en relación con el entorno, las rutas también se adaptan al entorno, ya que las decisiones del gestor de intersecciones afectan a la ruta generada.

Por otro lado, para lograr una adaptación más eficiente al entorno, se añaden variables para establecer prioridades de los AGV, las cuales se basan en diferentes indicadores comunes del entorno.

La adaptación al entorno es el factor más relevante en el *GA3*, por lo que se proponen mejoras como el aprendizaje simultáneo de todas las intersecciones, a diferencia del aprendizaje secuencial del *GA2*. También se incluyen datos de otras intersecciones para tomar decisiones en cada intersección.

Se propone un algoritmo generador de Gestores de flotas adaptados al entorno, que calcula continuamente la posición futura de los AGV teniendo en cuenta el tráfico y las decisiones del gestor de intersecciones. Esta metodología de cálculo de la posición futura se utiliza para determinar las rutas de los AGV. A través del aprendizaje, el gestor de intersecciones se adapta al entorno según los resultados del generador de rutas, generando así que las rutas se adapten al entorno. Concluiremos presentando el funcionamiento global de este generador de Gestor de flotas adaptados al entorno y su implementación práctica.

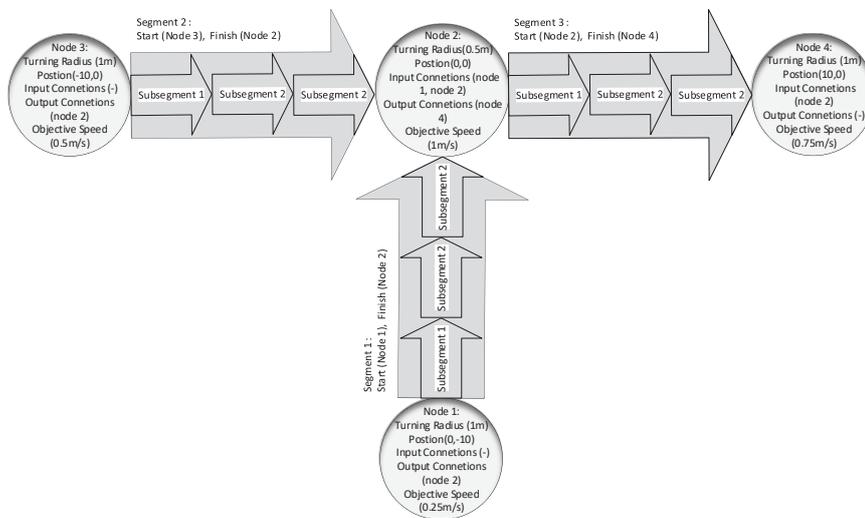
En esta Fase III no se expondrán los resultados de la evaluación del algoritmo *GA3*, ya que estos se encuentran comprendidos en el Capítulo 11. La exposición del algoritmo *GA3* está compuesta por los siguientes apartados:

- **Definiciones del entorno:** Se expondrá cómo se definen las diferentes partes del entorno en el algoritmo *GA3*.
- **Cálculo de la posición futura de los AGV:** Se presentarán las metodologías para el cálculo de la posición futura de los AGV, las cuales serán utilizadas por otros apartados del algoritmo.
- **Cálculo mejorado de rutas:** Se propondrá un sistema de cálculo de rutas de los AGV que incorpore el tráfico futuro en su cálculo.
- **Prioridad de los AGV:** Se incluirán prioridades en los AGV que serán utilizadas por el algoritmo para tomar decisiones.
- **Gestión de intersecciones:** Se fundamentará en la gestión de las intersecciones basada en *GA2*, pero con modificaciones realizadas con el objetivo de mejorar su rendimiento.
- **Algoritmo *GA3* completo:** Se expondrá la metodología de generación de Gestor de flotas adaptados al entorno y el funcionamiento del Gestor de flotas generado por *GA3*.

## 10.1 Definiciones del entorno

Antes de exponer el algoritmo propuesto, es necesario especificar cómo se van a definir las diferentes partes del entorno en el algoritmo. Para ello, se establecerán las definiciones de elementos como el *Roadmap*, los AGV, las tareas y algunas peculiaridades del algoritmo.

### 10.1.1 El Roadmap



**Figura 10.1.1:** Ejemplo de la representación del *Roadmap* en el algoritmo de la Fase III

El *Roadmap* para este algoritmo se compone de dos partes: los nodos y los segmentos. Los nodos representan las diferentes posiciones del mapa, que pueden ser destinos de rutas y también destinos iniciales, intermedios o finales de las tareas. Estos nodos están conectados mediante segmentos unidireccionales, que determinan las posibles direcciones de movimiento de los AGV. A su vez, los segmentos se dividen en Subsegmentos con un tamaño máximo definido. Este factor se utiliza para calcular rutas, considerando el tráfico estimado. Si analizamos los datos que definen un *Roadmap*, como se observa en el ejemplo de la Figura 10.1.1, son los siguientes:

- Por cada **nodo**:
  - Posición *X* y *Y* según las coordenadas del mapa.
  - Radio de giro: Determina el radio de la ruta que realizarán los AGV

- entre un segmento de entrada y otro de salida.
- Conexiones de entrada al nodo, determinadas por:
  - \* ID del segmento por el que entran los AGV
  - \* ID del nodo del que procede el segmento.
- Conexiones de salida al nodo, determinadas por:
  - \* ID del segmento por el que saldrán los AGV.
  - \* ID del nodo de destino del segmento.
- Velocidad máxima de circulación por el nodo.
- Por cada **segmento**:
  - ID del nodo de inicio del segmento.
  - ID del nodo de finalización del segmento.
  - Número de Subsegmentos que tiene el segmento.
  - Por cada **Subsegmento**: Tiempo necesario para cruzar por el Subsegmento a lo largo del tiempo, contando con un valor por cada una de las ventanas temporales. Está separado en dos partes: el tiempo para cruzar sin tráfico y el tiempo extra generado por el tráfico.

#### 10.1.1.0.1 Subsegmentos

En esta aproximación vemos como, en mapas grandes, un segmento puede tener una decena de metros, y diferentes valores del algoritmo se substraen de los segmentos. Con el objetivo de tener mayor resolución, proponemos la separación de los segmentos en subpartes mediante los Subsegmentos. A continuación, mostraremos la metodología de cálculo del número de Subsegmentos de un segmento mediante la fórmula 10.1.1:

$$NSubseg_j = \left\lceil \frac{\sqrt{(FX_j - SX_j)^2 + (FY_j - SY_j)^2}}{DMaxSubseg} \right\rceil \quad (10.1.1)$$

Donde:

- $NSubseg_j$ : Número de subsegmentos del segmento  $j$ .
- $FX_j$ : Posición del nodo final del segmento  $j$  en el eje  $X$ .
- $SX_j$ : Posición del nodo de inicio del segmento  $j$  en el eje  $X$ .
- $FY_j$ : Posición del nodo final del segmento  $j$  en el eje  $Y$ .
- $SY_j$ : Posición del nodo de inicio del segmento  $j$  en el eje  $Y$ .
- $DMaxSubseg$ : Distancia máxima que puede tener un subsegmento; idealmente, el tamaño del segmento tendría que ser, aproximadamente, el tamaño medio de todos los AGV.

#### 10.1.2 El AGV

El siguiente elemento a especificar es el AGV. En este apartado especificaremos los datos necesarios que determinan el comportamiento de los AGV. Estos son

los siguientes:

- Dinámica del AGV:
  - Velocidad máxima del AGV.
  - Aceleración angular máxima del AGV.
  - Aceleración del AGV.
  - Desaceleración del AGV.
- Dimensiones del AGV.
- Dimensiones y posición de detección del escáner de seguridad delantero del AGV. (Comportamiento para el cumplimiento de la normativa *ISO 3691-4* [19]).
- Dimensiones y posición de detección del escáner de seguridad trasero del AGV. (Comportamiento para el cumplimiento de la normativa *ISO 3691-4* [19]).
- Tiempo necesario para arrancar el AGV. (Comportamiento para el cumplimiento de la normativa *ISO 3691-4* [19]).

### 10.1.3 Rutas

La definición de la ruta es muy relevante a la hora de calcularla, ya que determina el método de extracción de la misma y la metodología que deben seguir los AGV para llegar a un destino. Algunos de los parámetros relacionados con la definición de la ruta son los siguientes:

1. ID de la misión.
2. ID de la tarea.
3. ID del AGV asignado.
4. Número de destinos.
5. Destinos:
  - (a) ID del destino.
  - (b) Si el AGV se tiene que detener, o no, en este destino.
  - (c) Tarea que se tiene que realizar en el nodo:
    - i. Carga o descarga de los elementos a transportar.
    - ii. Tiempo de espera en el punto.
    - iii. Características del elemento a transportar.
    - iv. Carga de baterías.
  - (d) Distancia recorrida.
  - (e) Tiempo de llegada.
  - (f) La curva a realizar en el destino:
    - i. Tipo de movimiento. Tomar una curva o rotar en el destino sin movimiento en los ejes  $X$  y  $Y$ .
    - ii. Velocidad de entrada a la curva (teniendo en cuenta la velocidad del destino anterior y la velocidad máxima del AGV) que tiene que seguir el vehículo en esta curva).

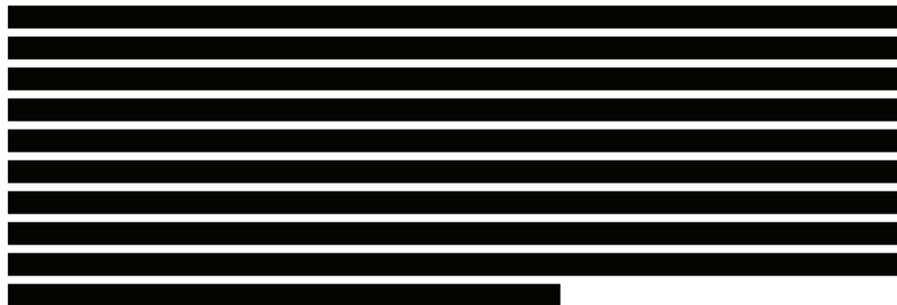
- iii. Velocidad de salida (teniendo en cuenta la velocidad del destino y la velocidad máxima del AGV) que tiene que seguir el vehículo en esta curva).
  - iv. Velocidad de la curva (teniendo en cuenta la velocidad del destino, la velocidad máxima del AGV y la aceleración angular máxima del AGV) comprendida dentro del radio de la curva del destino.
6. Segmento actual por el que se encuentra circulando el AGV.
  7. Segmento anterior por el que ha circulado el AGV.
  8. Próximo segmento por el que circulará el AGV.
  9. Último nodo que ha cruzado el AGV.
  10. Próximo nodo que cruzará el AGV.
  11. Tiempo de inicio de la misión.
  12. Tiempo objetivo de llegada al destino de la misión.
  13. Tiempo máximo permitido para llegar al destino de la misión.
  14. Tiempo restante estimado de llegada al destino de la misión.
  15. Tiempo estimado de llegada al destino de la misión.
  16. Tiempo final de llegada al destino de la misión.
  17. Tiempo final de trayecto del AGV para llegar al destino.
  18. Tiempo final de llegada respecto al tiempo máximo permitido.
  19. Tiempo final de llegada respecto al tiempo objetivo.

En los siguientes apartados, expondremos cómo se utilizan y calculan estos datos.

#### 10.1.4 Intersecciones

Otro factor relevante es cómo se definen las intersecciones; para estas nos basaremos en las definiciones utilizadas en la Fase I de la tesis, concretamente, queda definido en el apartado 8.3, al que añadiremos el ID de cada intersección.

## 10.2 Cálculo de la posición futura de los AGV



[Redacted text block]

10.2.1 Datos de inicialización

[Redacted text block]

[Redacted text block]

[Redacted text block]

### 10.2.2 Método de cálculo de la posición futura de los AGV

[Redacted text block]

[Redacted text block]

[Redacted text block]



*Figura* 10.2.1: Diagrama del cálculo de la posición futura de los AGV

[Redacted text block]

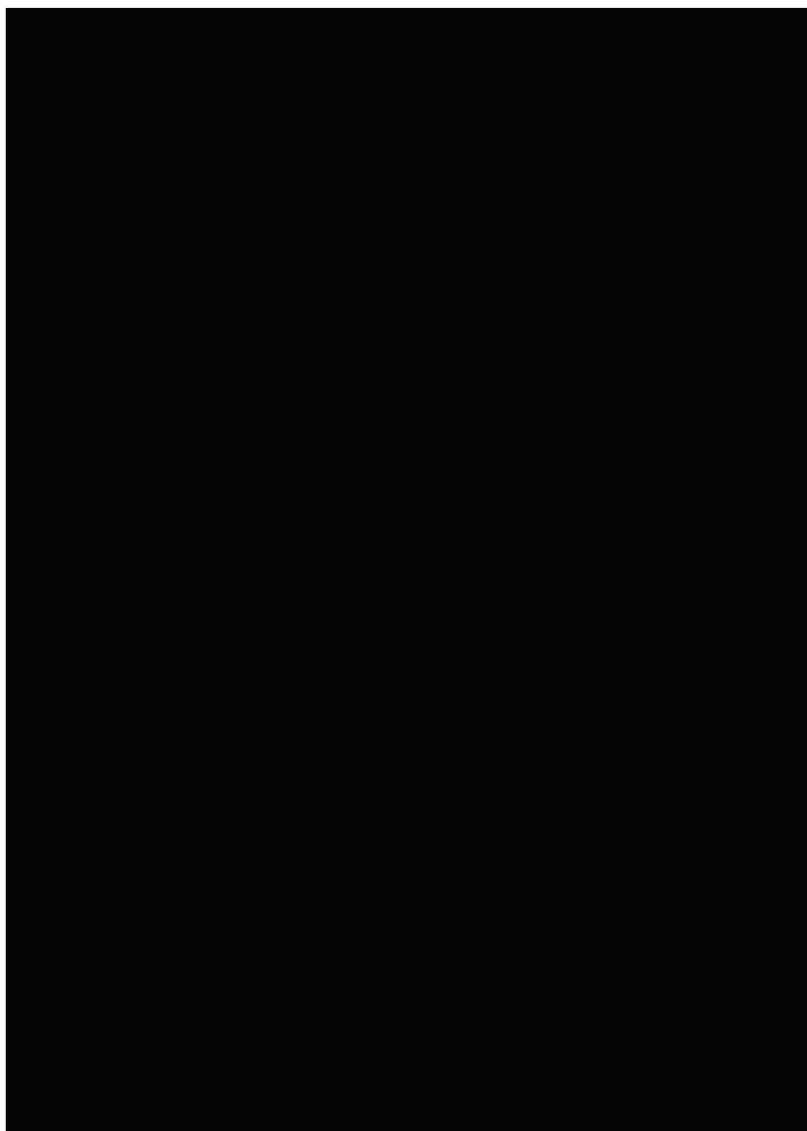
### 10.2.3 Cálculo dinámico de la movilidad de los AGV



*Figura 10.2.2:* Diagrama del algoritmo del movimiento dinámico de los AGV



[Redacted text block]



*Figura 10.2.3:* Diagrama del algoritmo de selección de zonas



[Redacted text block]



[Redacted text block]

10.2.3.0.1 Posición de los AGV

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]





[Redacted text block]

[Redacted text block]

#### 10.2.4 Emulación de la gestión del tráfico en las intersecciones

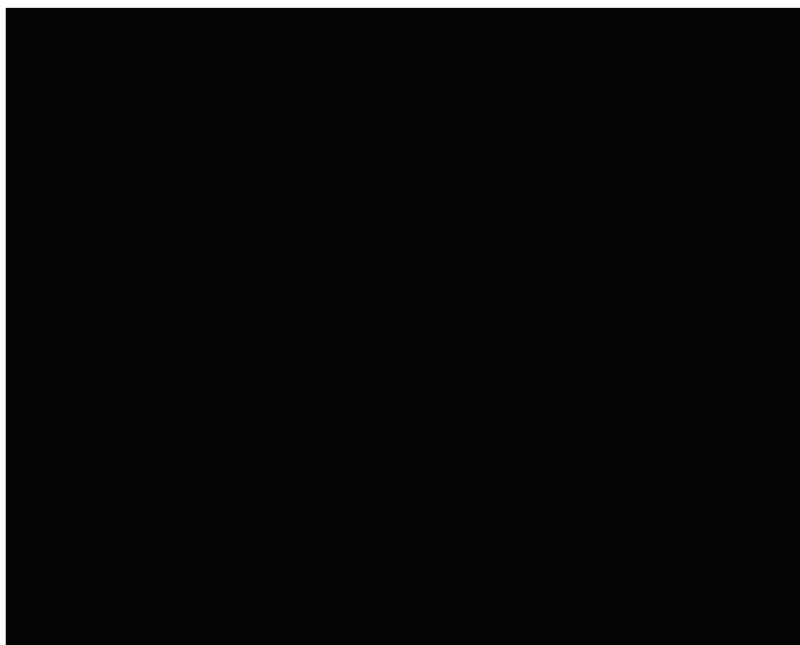
[Redacted text block]

#### 10.2.5 Detención causada por el tráfico

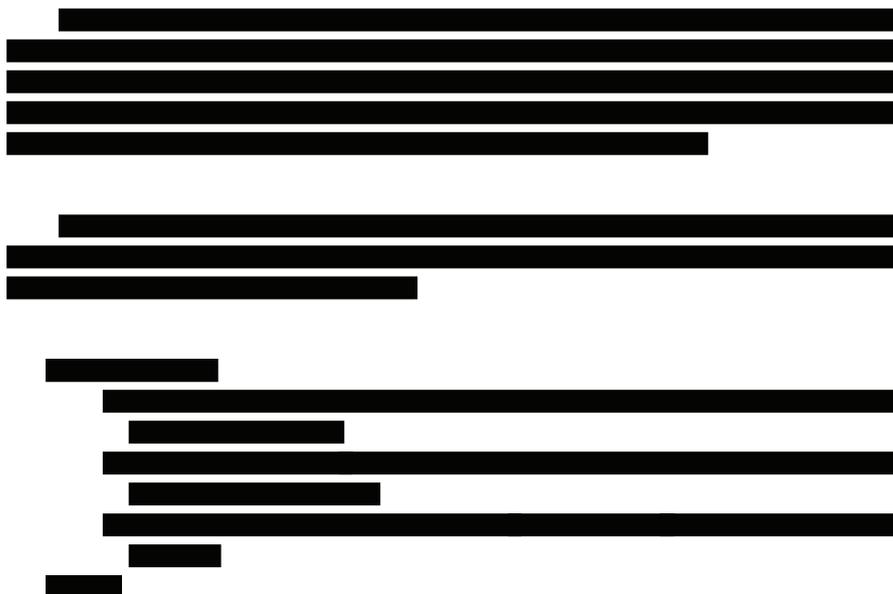
[Redacted text block]

##### 10.2.5.0.1 Propuesta mediante ocupación de una Red de Petri

[Redacted text block]



*Figura* 10.2.4: Ejemplo de la secuencia de activación de una Red de Petri



[Redacted text block]

[Redacted]

[Redacted]

[Redacted]

10.2.5.0.2 Detección perimetral

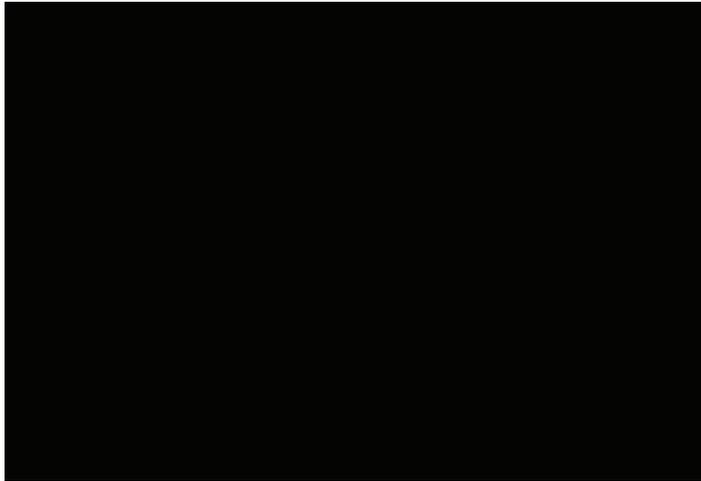
[Redacted]

[Redacted]

[Redacted]

[Redacted]

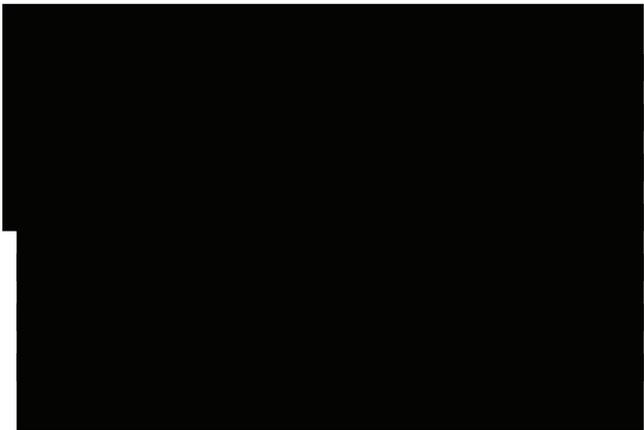
[Redacted text]



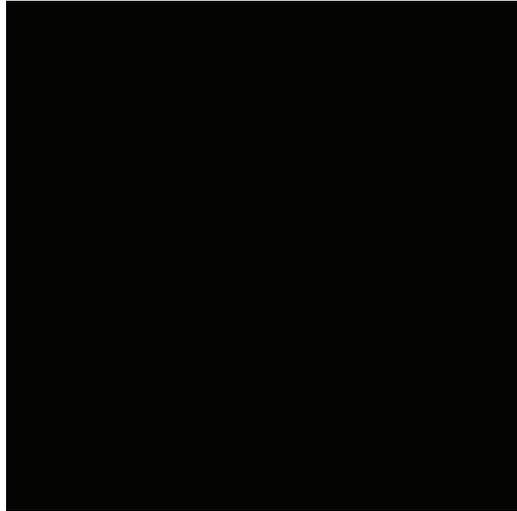
[Redacted text]

[Redacted text]

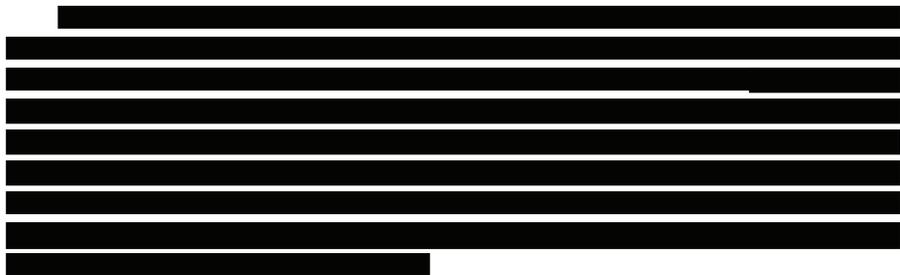
[Redacted text]

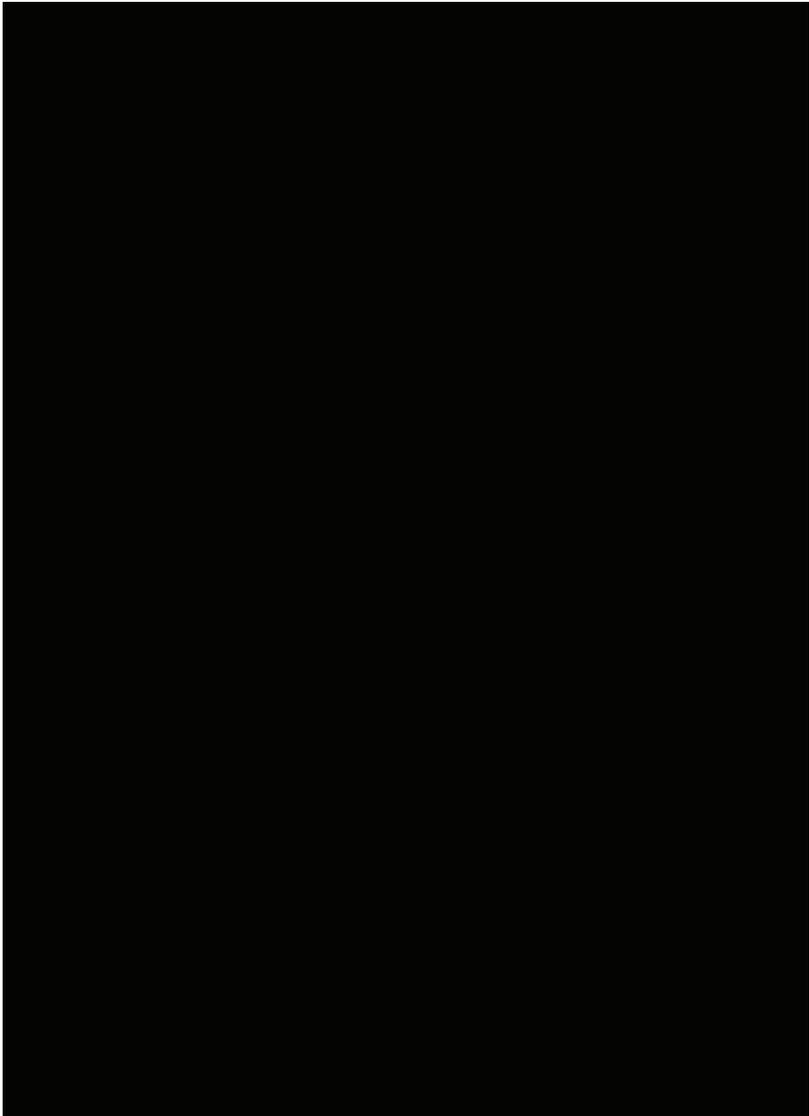






*Figura* 10.2.6: Representación de los puntos en un rectángulo





*Figura 10.2.7:* Diagrama del cálculo de la detección generada por el tráfico



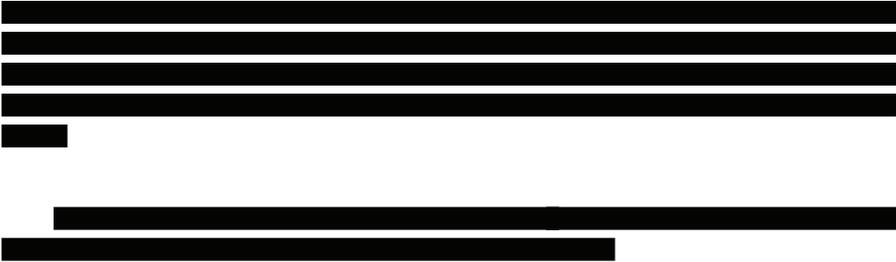
[Redacted text block]

ante el cálculo de detección por rectángulos del paso 5 para determinar

[Redacted text block]

**10.3 Cálculo mejorado de rutas**

[Redacted text block]



10.3.1 Generación del grafo

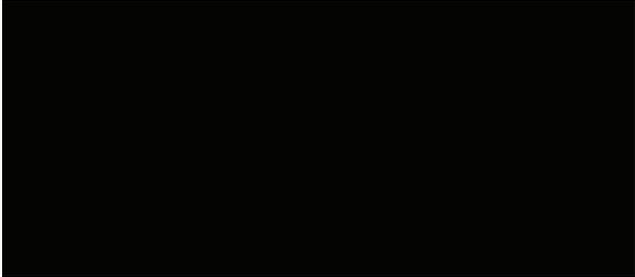
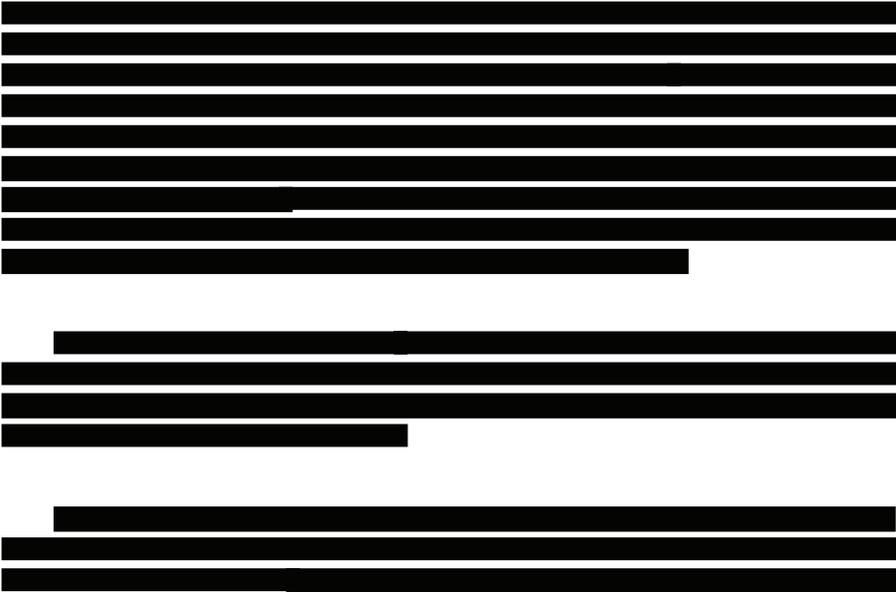
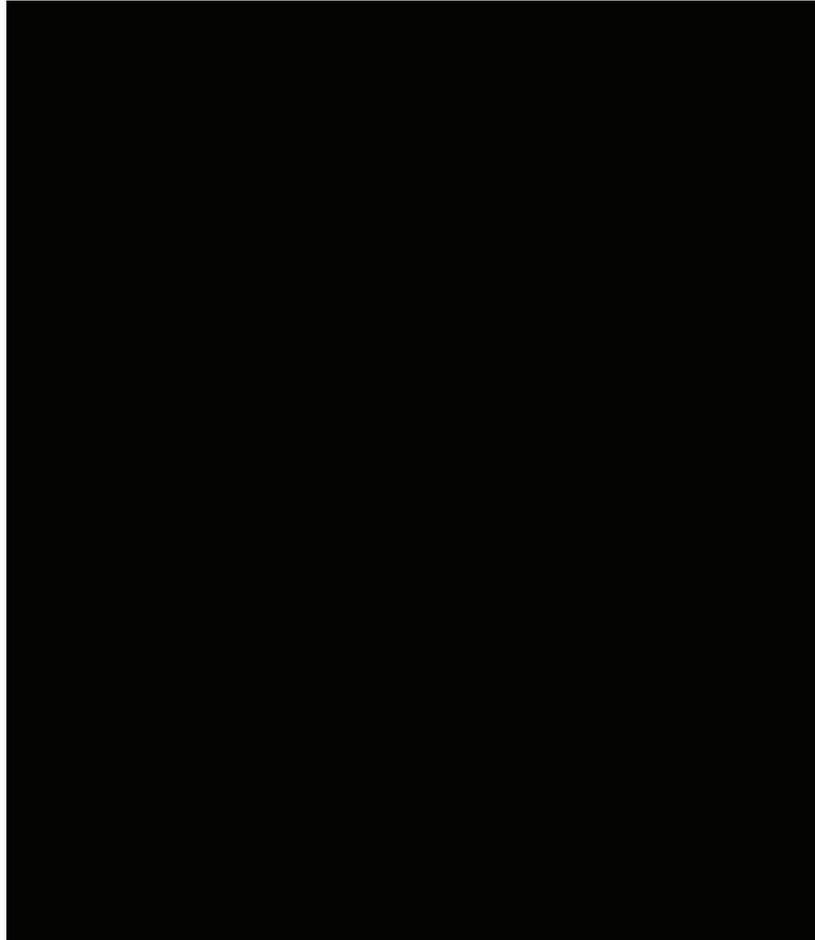


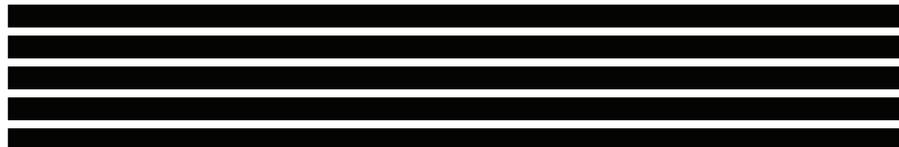
Figura 10.3.2: Diagrama ejemplo de un grafo del entorno para el cálculo de la ruta de los AGV



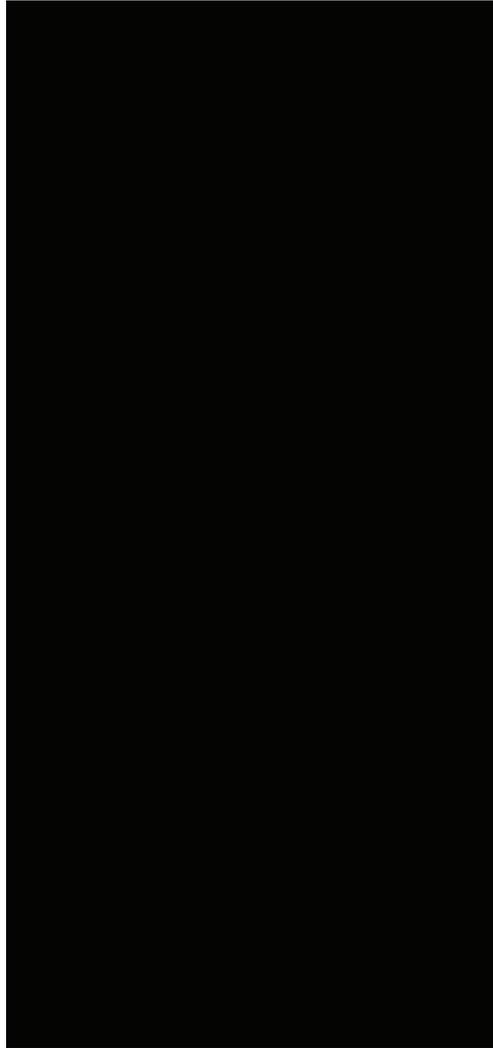
### 10.3.2 Resolución del grafo



*Figura* 10.3.3: Diagrama del algoritmo de resolución del grafo de la ruta



[Redacted text block]



*Figura* 10.3.4: Diagrama ejemplo de la resolución del grafo de la ruta



[Redacted text block]

### 10.3.3 Cálculo de un segmento

[Redacted text block]



[Redacted text block]

#### 10.3.4 Extracción del camino más corto

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]





*Figura* 10.3.5: Diagrama ejemplo de la resolución del grafo de la ruta



[Redacted text block]

10.3.5 Generación de la ruta

[Redacted text block]

[Redacted text block]

[Redacted]

## 10.4 Prioridad de los AGV

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

## 10.5 Gestión de intersecciones

[Redacted]

[Redacted]

[Redacted]

[Redacted]

### 10.5.1 Aprendizaje

[Redacted]

[Redacted]

[Redacted text block]

[Redacted text block]

[Redacted text block]

#### 10.5.1.0.1 Inicialización

[Redacted text block]

#### 10.5.1.0.2 Evaluación

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

**10.5.1.0.3 Mutación**

[Redacted text block]

**10.5.1.0.4 Generaciones**

[Redacted text block]

**10.5.1.0.5 Algoritmo de aprendizaje**

[Redacted text block]

[Redacted text block]

[Redacted text block]



*Figura 10.5.1:* Diagrama del algoritmo de aprendizaje de la Fase III



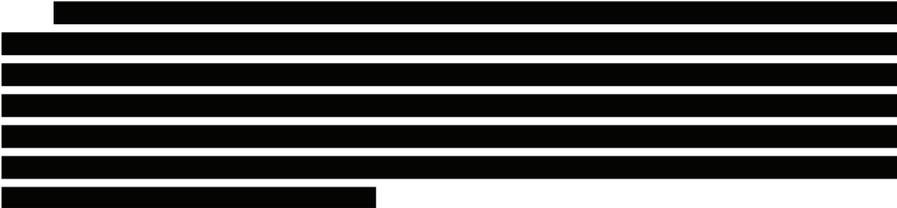
[Redacted text block]

10.5.2 Extracción de los datos

[Redacted text block]



*Figura 10.5.2:* Comparativa de los resultados con las diferentes propuestas de información a utilizar por una intersección de las demás intersecciones (Valor más alto mejor)



[Redacted text block]



[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

10.5.3 Fase de selección

[Redacted text block]

#### 10.5.4 Fase de comprobación

[Redacted text block]

#### 10.5.5 Algoritmo completo de gestión de intersecciones

[Redacted text block]

[Redacted text block]

[Redacted text block]



*Figura 10.5.3:* Diagrama del algoritmo de la gestión de las intersecciones del algoritmo GA3



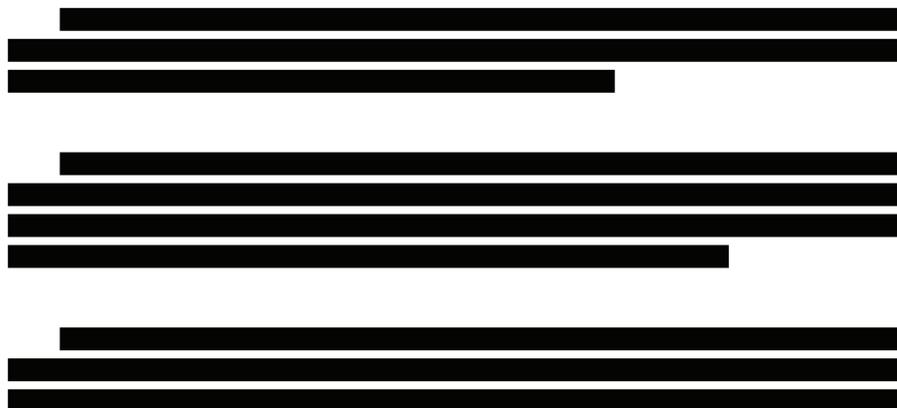
[Redacted text block]

10.6 Algoritmo GA3 completo

[Redacted text block]



*Figura* 10.6.1: Diagrama del proceso de generación de un gestor de flotas adaptado al entorno por parte del algoritmo *GA3*



[Redacted text block]



*Figura 10.6.2:* Funcionamiento global del algoritmo GA3

## Capítulo 11

# Evaluación del algoritmo propuesto

### Resumen

En este capítulo se presenta la evaluación de la versión más completa del algoritmo propuesto, el *GA3*. En primer lugar, se realiza una evaluación en los tres entornos propuestos y se comparan los resultados con los otros algoritmos de referencia. En segundo lugar, se exponen los resultados obtenidos en un entorno real. Con todos los resultados expuestos, se procede al análisis del resultado del *GA3* de todos los entornos, concluyendo que el algoritmo propuesto *GA3* consigue un mayor rendimiento en términos de reducción de costes respecto a los demás algoritmos de referencia. Finalmente, se realiza una prueba de hipótesis, donde se concluye, mediante esta metodología, que generar un algoritmo adaptado a un entorno específico puede reportar mejores resultados.

El capítulo se organiza en los siguientes apartados:

- **Resultados en los tres entornos simulados:** Se presenta el rendimiento del algoritmo *GA3* en los tres primeros entornos propuestos, incluyendo la carga computacional y el tiempo requerido para el aprendizaje. Además, se comparan estos resultados con los algoritmos de referencia y con el rendimiento teórico máximo.
- **Resultados en entorno de laboratorio real:** Se describe un caso de uso real

y se exponen los resultados obtenidos después de aplicar el algoritmo de Gestión de flotas generado por el *GA3*.

- **Análisis de los resultados:** Se analiza el resultado obtenido por el algoritmo *GA3* en comparación con los algoritmos de referencia.
- **Prueba de hipótesis:** Se realiza la prueba de hipótesis a la hipótesis de esta tesis.

## 11.1 Resultados en los tres entornos simulados

### 11.1.1 Entorno y cuantificación

Con el objetivo de evaluar y realizar el aprendizaje en diferentes entornos, analizaremos el aprendizaje y la evaluación del rendimiento del algoritmo en los tres primeros entornos propuestos en el Capítulo 6, con el criterio de evaluación requerido para cada entorno. Concretamente, estos son los siguientes:

1. **Entorno 1:** Expuesto en la Sección 6.2. Con el criterio de cuantificación *Cri1*.
2. **Entorno 2:** Expuesto en la Sección 6.3. Con el criterio de cuantificación *Cri2*.
3. **Entorno 3:** Expuesto en la Sección 6.4. Con el criterio de cuantificación *Cri3*.

### 11.1.2 Algoritmos de referencia

Para tener una referencia de rendimiento, se comparará el desempeño obtenido por el *GA3* con los algoritmos de referencia previamente expuestos en el Capítulo 5. Además, en el Entorno 3, se realizará una comparación adicional con el algoritmo propuesto *GA2*. Los algoritmos de Gestión de flotas utilizados como referencia en este capítulo son los siguientes:

- *STPRAN*
- *STPFIFO*
- *STPACC*
- *STPCT*
- *GEOPAR*
- *ASTAR*
- *TIMEWINDOW*
- *GA2*

### 11.1.3 Comparativa entre algoritmos en el entorno 1

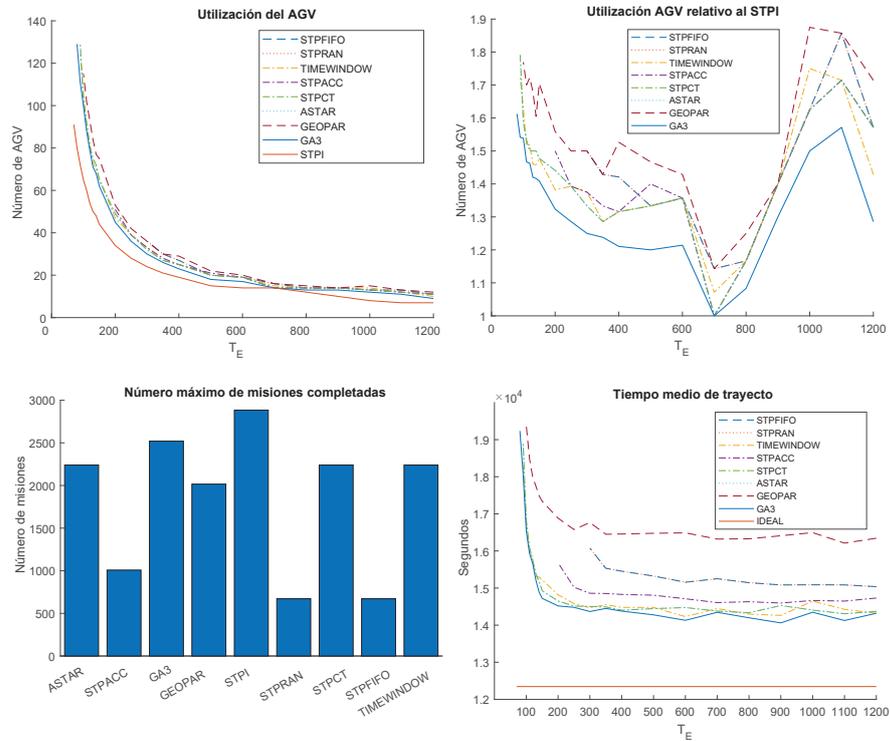


Figura 11.1.1: Resultados de la Fase III en el entorno 1

Este primer entorno tiene la peculiaridad de que, en lo referente al cálculo de rutas, tiene una relevancia baja, ya que existen pocos caminos óptimos para ir de una localización a otra. Solo en algunos casos con mucho tráfico existe una ruta alternativa que se convierte en la más óptima. Como el algoritmo propuesto calcula las rutas teniendo en cuenta el retraso futuro mediante la estimación de la interacción futura entre vehículos, en este entorno es capaz de encontrar el camino más óptimo en las situaciones más críticas con un alto nivel de tráfico. Pero, en este entorno, el factor más relevante es la gestión del tráfico en las dos intersecciones de entrada y salida del almacén. Por lo que, al estar adaptado a él, es capaz de lograr una mayor cantidad de paso de AGV que el resto de algoritmos. Estos dos factores logran que, en este entorno, el algoritmo propuesto sea capaz de conseguir un mejor rendimiento que el resto de algoritmos, como se puede observar en las gráficas de resultados de la Figura 11.1.1.

Si nos centramos en el resultado más relevante para evaluar este entorno: el número máximo de misiones completadas en el tiempo especificado situado en la gráfica inferior izquierda de la Figura 11.1.1, observamos que el algoritmo propuesto es capaz de lograr un sustancialmente mayor número de misiones completadas, con un rendimiento aproximadamente un 10 % superior al resto de los mejores algoritmos de referencia.

En cuanto a la media del tiempo de desplazamiento, representada en la gráfica inferior derecha de la Figura 11.1.1, y el número de AGV utilizados, mostrados en las dos gráficas superiores de la misma Figura 11.1.1, se observa una leve mejora en el tiempo de desplazamiento en comparación con los mejores resultados de los otros algoritmos. Sin embargo, esta mejora es inferior al incremento en el número de AGV utilizados. En este entorno, el número de AGV utilizados aumenta cuando no se puede seleccionar un vehículo que ya haya sido asignado previamente al comienzo de una tarea. El algoritmo propuesto tiene dos factores que contribuyen a reducir este valor: en primer lugar, al contar con un tiempo objetivo de llegada, se minimizan los vehículos con mayor retraso, lo que reduce la necesidad de vehículos adicionales; en segundo lugar, se penaliza el valor tiempo de trayecto medio en beneficio de minimizar el número de vehículos utilizados.

### 11.1.4 Comparativa entre algoritmos en el entorno 2

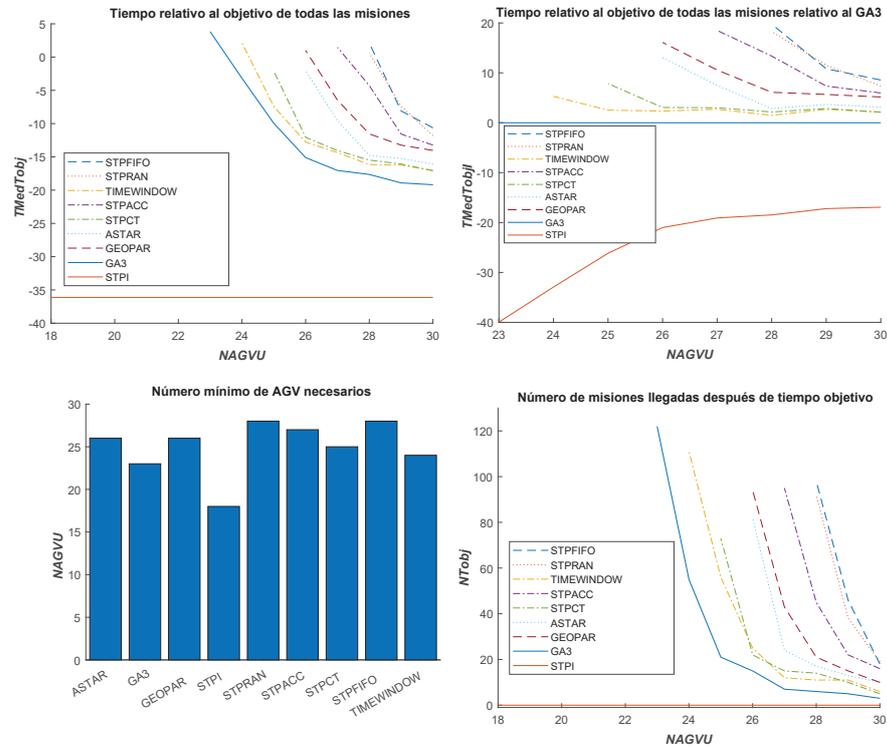


Figura 11.1.2: Resultados de la Fase III en el entorno 2

El segundo entorno es bastante diferente al anterior, ya que al calcular las rutas, en muchas ocasiones solo existe una ruta posible. Solo en pocos casos hay otras rutas alternativas. Es por ello que en este entorno el cálculo de rutas es muy poco relevante. Las intersecciones son sencillas y cuentan con pocas entradas y salidas, pero al mismo tiempo el número de cruces es muy elevado. En algunos casos, es posible agruparlos en cruces más complejos, pudiendo así explotar la gestión de las intersecciones del GA3. El objetivo de aprendizaje de este algoritmo será maximizar el paso de los AGV por las intersecciones, lo que reporta un mayor rendimiento que los demás algoritmos.

Si analizamos el rendimiento del algoritmo GA3 en el segundo entorno, como se muestra en la Figura 11.1.2, podemos observar que ha obtenido el mejor resultado en la variable más relevante de este entorno: el número mínimo de AGV necesarios para completar las tareas. Esto se refleja en la gráfica inferior izquierda de la Figura 11.1.2, donde se observa que se requieren 23 AGV, uno

menos que el mejor resultado obtenido por los otros algoritmos en este entorno, específicamente el algoritmo *TIMEWINDOW* con 24 AGV.

Por otro lado, si analizamos el resto de parámetros relevantes para determinar el rendimiento del algoritmo *GA3*, como se muestra en la Figura 11.1.2, podemos observar que supera a los demás algoritmos. En la gráfica inferior derecha se representa el número de veces que los AGV llegaron después del tiempo objetivo, y en las dos gráficas superiores se muestra la media de este tiempo. En ambos casos, el algoritmo propuesto obtiene mejores resultados que los demás algoritmos. Esto demuestra que la gestión de los cruces es una fortaleza de este algoritmo.

### 11.1.5 Comparativa entre algoritmos en el entorno 3

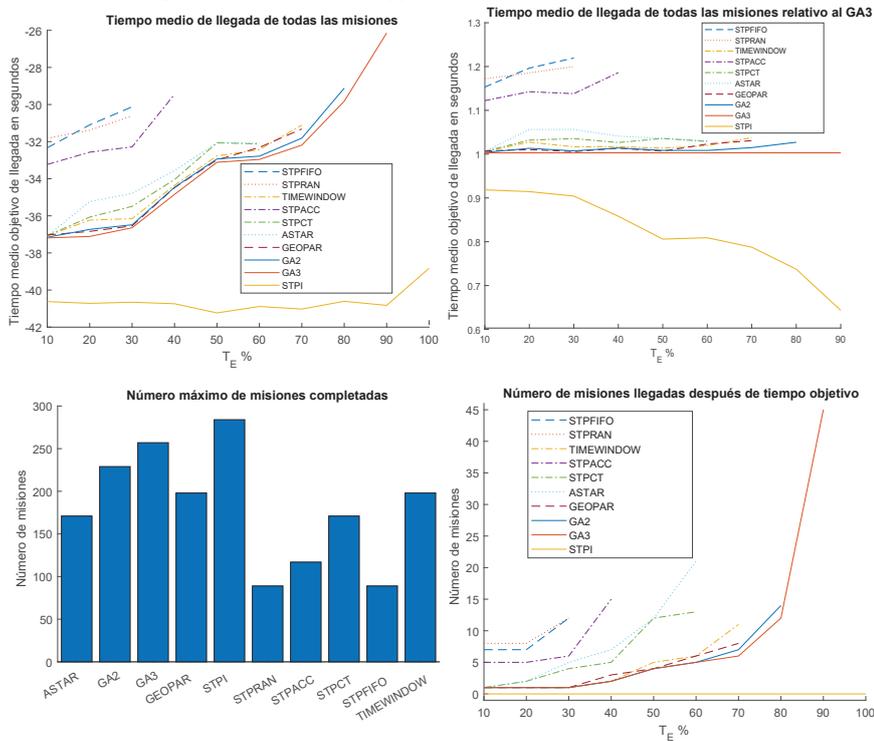


Figura 11.1.3: Resultados de la Fase III en el entorno 3

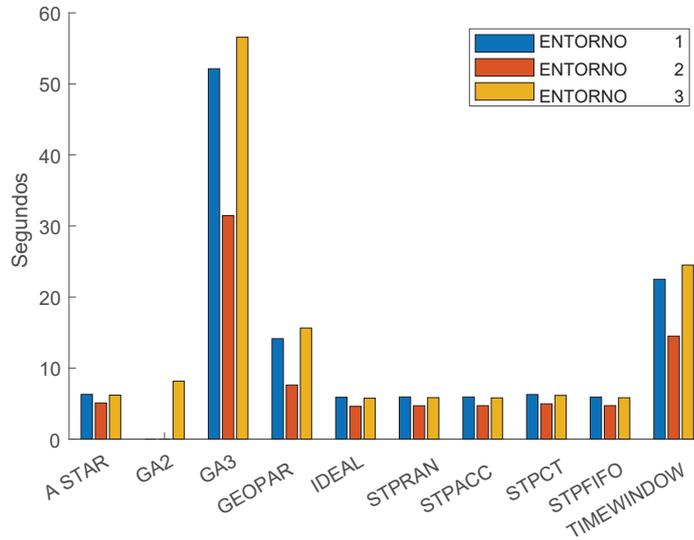
Este tercer entorno, debido a su configuración y distribución, permite una variabilidad de rutas. por lo que podemos observar su rendimiento en el cálculo de rutas con consideración del tráfico. Por otro lado, en este entorno también

es relevante la gestión de las intersecciones, al contar con un valor elevado con multitud de entradas y salidas.

Si analizamos el rendimiento del algoritmo en el entorno mediante la Figura 11.1.3, observamos que, con todas las mejoras realizadas, el algoritmo *GA3* es capaz de arrojar unos resultados satisfactorios con un porcentaje de *PST* del 90 %, logrando un máximo de misiones completadas de 257. Esto aumenta sustancialmente el rendimiento respecto al *GA2*, que es capaz de lograr un valor del 80 %. Analizando otro valor significativo: el número de misiones que han llegado a cada destino después de *Tobj*, como se muestra en la gráfica inferior derecha de la Figura 11.1.3, observamos que el valor de esta nueva propuesta tiene un rendimiento similar al algoritmo *GA2*, en los resultados con un valor de *Te* respecto al *STPI* de hasta el 60 %. Después de este porcentaje se empieza a observar una pequeña diferencia, obteniendo el algoritmo propuesto *GA3* un mejor valor. Respecto a los otros dos algoritmos de referencia con mayor rendimiento: *TIMEWINDOW* y *GEOPAR*, tienen un rendimiento similar hasta un valor de *PST* del 30 %, obteniendo estos dos algoritmos un rendimiento ligeramente menor respecto al propuesto, con valores de *PST* por encima del 30 %. Los demás algoritmos obtienen un rendimiento sustancialmente inferior.

Por otro lado, en relación con la media de tiempo de llegada de las tareas a sus destinos respecto al *Tobj*, que puede observarse en las dos gráficas superiores de la Figura 11.1.3, obtenemos una ligera mejora respecto a los demás algoritmos, pero esta no es tan elevada como con el número de misiones completas. Esto denota que la priorización de los AGV logra minimizar los peores casos.

### 11.1.6 Carga computacional



**Figura 11.1.4:** Carga computacional de los diferentes entornos en la Fase III (Valor más bajo mejor)

El incremento de la complejidad del algoritmo suele conllevar que se requiera una mayor carga computacional. Para comprobar este factor, recogeremos la carga computacional en los tres entornos mediante la Figura 11.1.4. Esta gráfica es la misma que las indicadas en las Figuras 7.2 y 9.4.2, añadiendo el algoritmo *GA3*.

Si analizamos la carga computacional del algoritmo propuesto *GA3*, observamos que es el algoritmo con una mayor carga computacional, teniendo un incremento de la carga computación que oscila entre un 600 % y un 900 % respecto al *STPI*. Existen tres procesos que incrementan, en gran medida, la carga computacional. Si los ordenamos en orden decreciente, son los siguientes:

1. Cálculo de la posición futura de los AGV: Requiere emular el comportamiento de los vehículos periódicamente generando una gran carga de trabajo.
2. Adquisición de datos para la toma de decisiones en las intersecciones: Este proceso, al contar con muchos datos, genera una carga computacional que, aunque no es muy elevada, al realizarse también en el cálculo de posición

futura, multiplica la carga relativa a este factor.

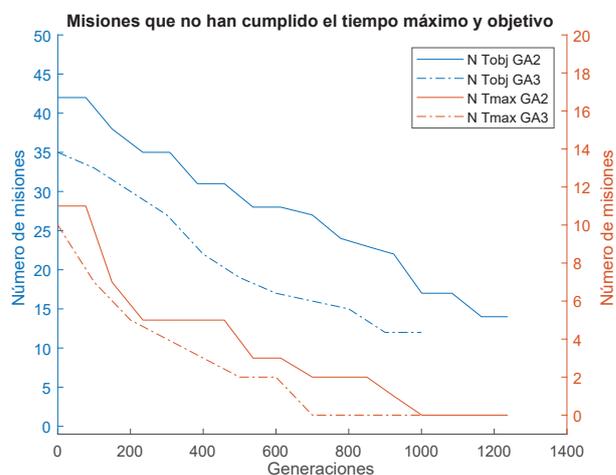
3. Cálculo de rutas: Al tener en cuenta el tráfico, aumenta la carga computacional por dos factores: el primero por tener que incluir en cada una de las ventanas temporales el tiempo extra generado por el tráfico futuro. El segundo factor se genera por la cantidad de cálculos necesarios para calcular la ruta.

Por lo tanto, este algoritmo propuesto requiere un procesamiento de cálculo más elevado que los algoritmos de referencia. Esto genera un incremento de costes, a tener en cuenta en escenarios poco exigentes y con poca cantidad de vehículos. Pero no supone un incremento significativo en los casos de uso expuestos.

Aunque el algoritmo propuesto, *GA3*, presenta una demanda computacional superior a la de los demás algoritmos, tal rendimiento no impide su operatividad con el hardware estándar que se ha empleado para el ensayo. Incluso en el peor de los escenarios, la simulación ha demostrado una velocidad 100 veces superior al tiempo real, lo que descarta problemas en su implementación. Con base en estas observaciones, se decide excluir este parámetro en la prueba de hipótesis, ya que el incremento de costes es despreciable comparado con los demás parámetros.

Para ilustrar que el incremento de costos asociado a la demanda computacional del algoritmo propuesto resulta despreciable en comparación con otros parámetros, consideremos el hardware necesario para las simulaciones. Con un costo de 3000 euros, este se torna mínimo cuando se le compara con el costo promedio de adquisición de los AGV expuestos, el cual asciende a 30000 euros. Teniendo en cuenta que sólo se necesita un hardware para ejecutar el Gestor de flotas y que la cantidad de vehículos puede variar, se concluye que el costo adicional generado por la demanda computacional es, en definitiva, despreciable.

### 11.1.6.1 Aprendizaje



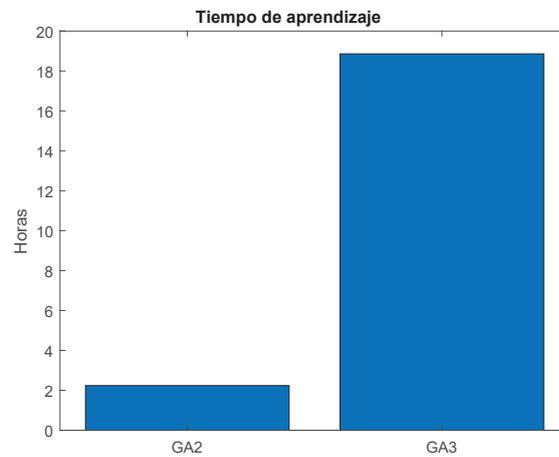
**Figura 11.1.5:** Resultados durante el aprendizaje de *GA2* y *GA3* en el Entorno 3, con valor de *PST* del 80 % (Valor más bajo mejor)

Con el objetivo de observar el proceso de aprendizaje del algoritmo *GA3* respecto al *GA2*, se procede a comparar el proceso de aprendizaje del tercer entorno, con un valor de *PST* del 80 %.

En la Figura 11.1.5 observamos el número de destinos a los que ha llegado un vehículo después del tiempo *Tmax* y *Tobj*, por cada una de las generaciones del proceso de aprendizaje. Como el aprendizaje del *GA2* es secuencial, para tener un valor de referencia se indicará el número de generaciones acumulado por el aprendizaje secuencial.

Inicialmente, en la primera generación observamos que el *GA3*, otorga un rendimiento ligeramente superior al *GA2*. Esto es debido al método mejorado de generación de *Genes* iniciales. Y también por las mejoras del algoritmo fuera del apartado de aprendizaje.

Al ir incrementando las generaciones, observamos una diferencia sustancial en la rampa de crecimiento de los dos algoritmos, debido al método de aprendizaje utilizado: el secuencial del *GA2* y el simultáneo del *GA3*. Como se observa en el método secuencial, en cada secuencia de aprendizaje un cruce se adapta al entorno. Como existen cruces que tienen mayor o menor relevancia, el incremento de aprendizaje por cada una de las intersecciones no es constante. Por otro lado, el aprendizaje simultáneo se caracteriza por ir reduciendo las



**Figura 11.1.6:** Tiempo de aprendizaje de *GA2* y *GA3* (Valor más bajo mejor)

mejoras por cada generación al ir incrementando el número de generaciones.

Otro factor relevante consiste en el tiempo requerido para realizar el aprendizaje. Si analizamos el tiempo requerido para realizar el aprendizaje de esta sección, y lo cuantificamos mediante la Figura 11.1.6, aunque el número de generaciones requeridas para el aprendizaje del *GA3* sea inferior al de *GA2*, la carga computacional del *GA3* es muy superior, necesitando, para este entorno y tareas, un tiempo total de 18 horas, mientras que el *GA2* ha requerido solo 2 horas. Este periodo requerido de 18 horas es asumible, no supone un incremento significativo en los costes por lo que es un coste asumible.

## 11.2 Resultados en entorno de laboratorio real

Una vez evaluado el algoritmo propuesto en los tres entornos planteados, se procede a implementar el algoritmo en el entorno de laboratorio real, descrito en la Sección 6.5. Para ello, se realizarán las tareas indicadas para diferentes algoritmos durante un periodo continuo de 4 horas, contando con 2 vehículos disponibles.

Dado que se trata de un entorno real, se han desarrollado únicamente los siguientes algoritmos de Gestión de flotas: *STPRAN*, *STPCT* y el algoritmo generado por el *GA3*. Por lo tanto, en la evaluación del rendimiento se compara únicamente respecto a estos tres algoritmos.

		STPRAN	STPCT	GA3
Tiempo de viaje	$T_{medV}$	44.07	43.38	42.16
	$T_{maxV}$	66.00	66.00	70.00
	$T_{medV}$	35.50	35.50	31.00
Tiempo diferencial con $T_{max}$	$T_{medT_{max}}$	-55.93	-56.62	-57.84
	$T_{maxT_{max}}$	-34.00	-34.00	-30.00
	$T_{minT_{max}}$	-64.50	-64.50	-69.00
Llegadas después de $T_{max}$	$NT_{max}$	0.00	0.00	0.00
Número de misiones	$NMISCo$	0.00	0.00	0.00
	$NotMISCo$	290.00	290.00	290.00

**Figura 11.2.1:** Resultados globales absolutos del entorno real

		STPRAN	STPCT	GA3
Tiempo de viaje	$T_{medV}$	0.0	1.6	4.3
	$T_{maxV}$	0.0	0.0	-6.1
	$T_{medV}$	0.0	0.0	12.7
Tiempo diferencial con $T_{max}$	$T_{medT_{max}}$	0.0	1.2	3.4
	$T_{maxT_{max}}$	0.0	0.0	-11.8
	$T_{minT_{max}}$	0.0	0.0	7.0
Llegadas después de $T_{max}$	$NT_{max}$	0.0	0.0	0.0
Número de misiones	$NMISCo$	0.0	0.0	0.0
	$NotMISCo$	0.0	0.0	0.0

**Figura 11.2.2:** Resultados globales respecto al STPRAN del entorno real

Una vez realizadas las pruebas, se presentan los resultados en las tablas de las Figuras 11.2.1 y 11.2.2. En estas tablas se pueden observar los resultados de los diferentes algoritmos empleando los criterios de evaluación propuestos, así como su rendimiento comparado con el STPRAN. Los parámetros mostrados en las tablas son los siguientes:

1.  $NMisCo$ . Número de misiones completadas en un entorno, cumpliendo los requisitos del mismo, en un periodo de tiempo específico.
2.  $NotNMisCo$ . Número de misiones no completadas en un entorno, en un periodo de tiempo específico.
3.  $NT_{max}$ . Número de veces que los AGV han llegado a sus destinos después del  $T_{max}$ .
4.  $TMedT_{max}$ . Tiempo medio de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
5.  $TMaxT_{max}$ . Tiempo máximo de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
6.  $TMinT_{max}$ . Tiempo mínimo de llegada de los AGV a sus diferentes destinos respecto a sus  $T_{max}$  respectivos. Medido en segundos.
7.  $TMedV$ . Tiempo medio de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.

8.  $TMaxV$ . Tiempo máximo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.
9.  $TMinV$ . Tiempo mínimo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos.

Al analizar el rendimiento de los indicadores según su importancia, se pueden extraer los siguientes resultados:

1. **Número de misiones completadas ( $NMisCo$ ) y número de misiones no completadas ( $NotNMisCo$ ):** Todos los algoritmos completan todas las tareas, por lo que se procede a evaluar el siguiente indicador.
2. **Llegadas después del  $Tmax$  ( $NT_{max}$ ):** Todos los algoritmos logran que todas las tareas lleguen antes de sus  $Tmax$  respectivos. Esto se debe a que la exigencia de esta prueba en este aspecto es baja, por lo que se procede a evaluar el siguiente indicador.
3. **Tiempo de viaje o trayecto de todas las tareas ( $TMedV$ ,  $TMaxV$  y  $TMinV$ ):** En este caso, se observa que  $STPCT$  mejora un 1.6% el tiempo de viaje medio respecto al  $STPRAN$ , sin alterar el valor  $TMaxV$ . Por otro lado, el  $GA3$  mejora un 4.3% alterando el valor  $TMaxV$ . Este indicador expone una mejora del rendimiento en el tiempo de viaje de estos dos algoritmos.

Por lo tanto, si consideramos los indicadores para este entorno, observamos que en este entorno real, el algoritmo  $STPRAN$ , aunque con un desempeño aceptable debido al entorno, ha obtenido el peor rendimiento. El algoritmo  $STPCT$  ha obtenido un mayor rendimiento, y el algoritmo generado por el  $GA3$  ha obtenido el mejor rendimiento. Se concluye que el algoritmo propuesto en esta tesis ha obtenido un mejor rendimiento que los algoritmos de comparación en este entorno.

Tarea ID	Tiempo de viaje en segundos								
	STPRAN			STPCT			GA3		
	TMedV	TMaxV	TMinV	TMedV	TMaxV	TMinV	TMedV	TMaxV	TMinV
T1_1-2	35.74	36.50	35.50	35.88	36.50	35.50	36.09	36.50	35.50
T1_2-1	44.09	44.50	43.50	44.18	44.50	43.50	44.12	44.50	43.50
T2_2-1	56.75	57.50	55.50	56.66	57.50	56.00	56.75	57.50	56.00
T2_1-3	36.66	37.00	36.00	36.66	37.00	36.00	36.72	37.00	36.00
T3_1-3	36.44	37.00	36.00	36.75	37.00	36.50	36.78	37.00	36.50
T3_3-1	50.00	51.00	49.50	37.50	38.00	37.00	37.41	38.00	37.00
T4_3-2	63.78	64.50	63.00	63.91	65.00	63.50	47.52	64.50	31.00
T4_1-3	36.53	37.00	36.00	36.50	37.00	36.00	36.56	37.00	36.00
T5_2-3	40.91	41.50	40.50	40.91	41.50	40.50	40.91	41.50	40.50
T5_3-1	37.37	38.00	37.00	37.44	38.00	37.00	37.44	38.00	37.00
T6_3-1	37.59	38.00	37.00	37.38	38.00	37.00	37.47	38.00	37.00
T6_1-2	36.03	36.50	35.50	36.00	36.50	35.50	35.97	36.50	35.50
T7_1-2	36.00	36.50	35.50	35.88	36.50	35.50	35.94	36.50	35.50
T7_2-3	40.97	41.50	40.50	40.91	41.50	40.50	41.00	41.50	40.50
T8_2-3	40.97	41.50	40.50	41.06	41.50	40.50	41.03	41.50	40.50
T8_3-2	49.25	49.50	48.50	49.16	49.50	48.50	49.25	49.50	48.50
T9_3-2	49.28	49.50	49.00	49.25	49.50	49.00	50.42	70.00	31.00
T9_2-1	65.38	66.00	64.50	65.28	66.00	65.00	44.13	44.50	43.50

Figura 11.2.3: Resultados de tiempo de viaje absolutos en el entorno real

Tarea ID	Tiempo de viaje en %								
	STPRAN			STPCT			GA3		
	TMedV	TMaxV	TMinV	TMedV	TMaxV	TMinV	TMedV	TMaxV	TMinV
T1_1-2	0.00	0.00	0.00	-0.41	0.00	0	-0.99	0.00	0.00
T1_2-1	0.00	0.00	0.00	-0.20	0.00	0	-0.07	0.00	0.00
T2_2-1	0.00	0.00	0.00	0.17	0.00	-0,9009	0.00	0.00	-0.90
T2_1-3	0.00	0.00	0.00	0.00	0.00	0	-0.17	0.00	0.00
T3_1-3	0.00	0.00	0.00	-0.86	0.00	-1,38889	-0.94	0.00	-1.39
T3_3-1	0.00	0.00	0.00	25.00	25.49	25,2525	25.19	25.49	25.25
T4_3-2	0.00	0.00	0.00	-0.20	-0.78	-0,79365	25.50	0.00	50.79
T4_1-3	0.00	0.00	0.00	0.09	0.00	0	-0.09	0.00	0.00
T5_2-3	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00
T5_3-1	0.00	0.00	0.00	-0.17	0.00	0	-0.17	0.00	0.00
T6_3-1	0.00	0.00	0.00	0.58	0.00	0	0.33	0.00	0.00
T6_1-2	0.00	0.00	0.00	0.09	0.00	0	0.17	0.00	0.00
T7_1-2	0.00	0.00	0.00	0.35	0.00	0	0.17	0.00	0.00
T7_2-3	0.00	0.00	0.00	0.15	0.00	0	-0.08	0.00	0.00
T8_2-3	0.00	0.00	0.00	-0.23	0.00	0	-0.15	0.00	0.00
T8_3-2	0.00	0.00	0.00	0.19	0.00	0	0.00	0.00	0.00
T9_3-2	0.00	0.00	0.00	0.06	0.00	0	-2.31	41.41	36.73
T9_2-1	0.00	0.00	0.00	0.14	0.00	-0,77519	32.50	32.58	32.56

Figura 11.2.4: Resultados de tiempo de viaje respecto al STPRAN en el entorno real

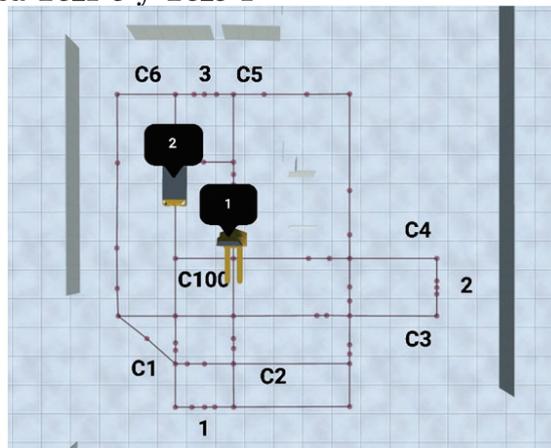
Una vez analizados los resultados globales, se procede a observar los datos detallados para identificar dónde se encuentran las diferencias de rendimiento. Si examinamos el tiempo de viaje de todas las tareas para cada algoritmo, mostradas en la Figura 11.2.3, y el mismo tiempo respecto a STPRAN en la Figura 11.2.4, podemos observar dónde se encuentran las diferencias entre los algoritmos. Se ha observado que, al ser un circuito reducido con pocos vehículos y pocas rutas posibles, en muchas tareas todos los algoritmos han tomado

la misma ruta, por lo tanto, cuentan con el mismo tiempo de trayecto. Al centrarnos en aquellas donde se constata una diferencia, mediante los datos extraídos del movimiento de los AGV y visualizándolos mediante la representación 3D, observamos las tareas donde existe una diferencia de rendimiento. El conjunto de tareas relevantes son las siguientes:

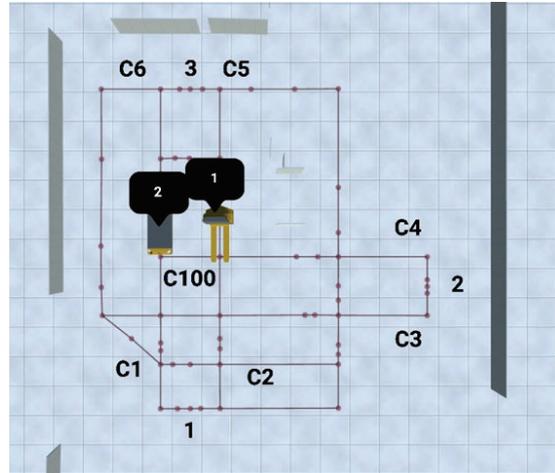
- Tarea T3\_1-3 y T3\_3-1
- Tarea T4\_3-2 y T4\_1-3
- Tarea T9\_3-2 y T9\_2-1

A continuación, se expondrá un análisis de las acciones realizadas por cada algoritmo en cada uno de los conjuntos de tareas expuestos en la lista anterior.

### 11.2.1 Tarea T3\_1-3 y T3\_3-1



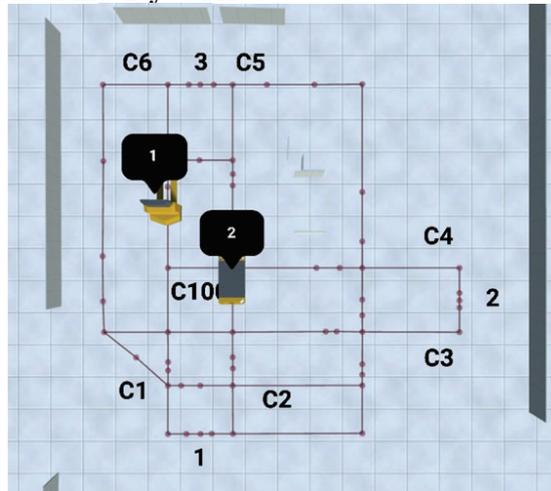
**Figura 11.2.5:** Visualización del movimiento de los AGV durante la ejecución de las tareas T3\_1-3 y T3\_3-1, utilizando el algoritmo *STPRAN* en el entorno real



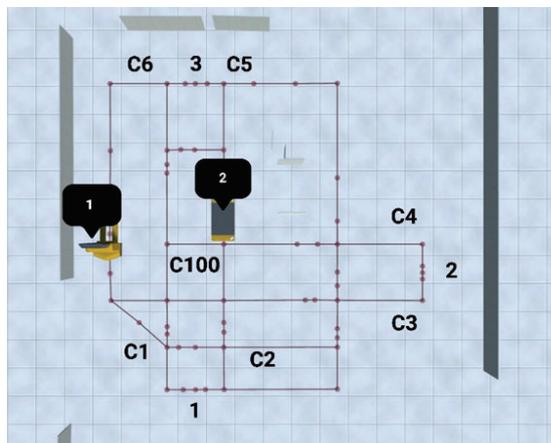
**Figura 11.2.6:** Visualización del movimiento de los AGV durante la ejecución de las tareas T3.1-3 y T3.3-1, utilizando los algoritmos *STPCT* o *GA3* en el entorno real

Como se observa en la Figura 11.2.5, el algoritmo *STPRAN* detiene el AGV 2, debido a que el AGV 1 ya ha entrado en la intersección. Este hecho se puede optimizar, ya que, por las rutas que tienen, no causarían colisión. Los algoritmos *STPCT* y *GA3* son capaces de dejar pasar a los 2 AGV, ya que determinan que, por sus rutas, estos no colisionarán. Esto se puede observar en la Figura 11.2.6.

### 11.2.2 Tareas T4.3-2 y T4.1-3



*Figura 11.2.7:* Visualización del movimiento de los AGV durante la ejecución de las tareas T4.3-2 y T4.1-3, utilizando los algoritmos *STPCT* o *STPRAN* en el entorno real



*Figura 11.2.8:* Visualización del movimiento de los AGV durante la ejecución de las tareas T4.3-2 y T4.1-3, utilizando el algoritmo *GA3* en el entorno real

En estas tareas, el algoritmo *STPRAN* y *STPCT* toman el camino más corto para el AGV 1, como se observa en la Figura 11.2.7. Sin embargo, como se observa en la Figura 11.2.8, el *GA3* decide que el AGV 1 tome el camino de la izquierda, optimizando el tiempo del AGV 1. Ese camino, con esta condición de

tráfico, tiene un tiempo de llegada menor, aunque tenga un trayecto más largo.

### 11.2.3 Tareas T9\_3-2 y T9\_2-1

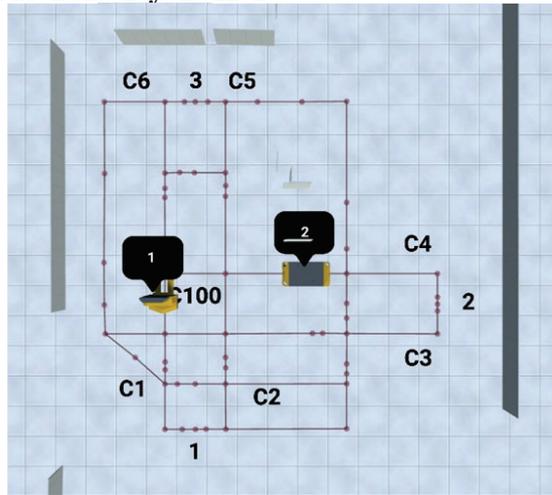


Figura 11.2.9: Visualización del movimiento de los AGV durante la ejecución de las tareas T9\_3-2 y T9\_2-1, utilizando los algoritmos *STPRAN* o *STPCT* en el entorno real

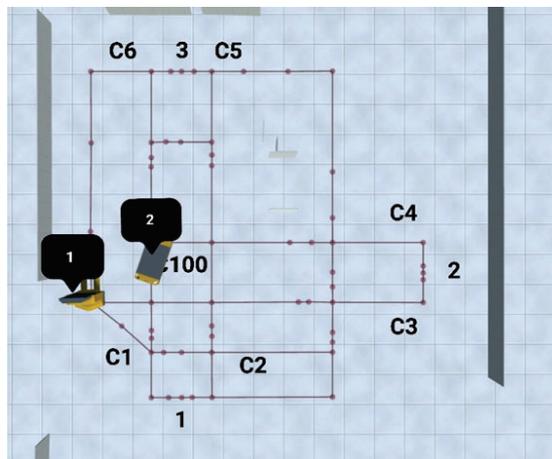
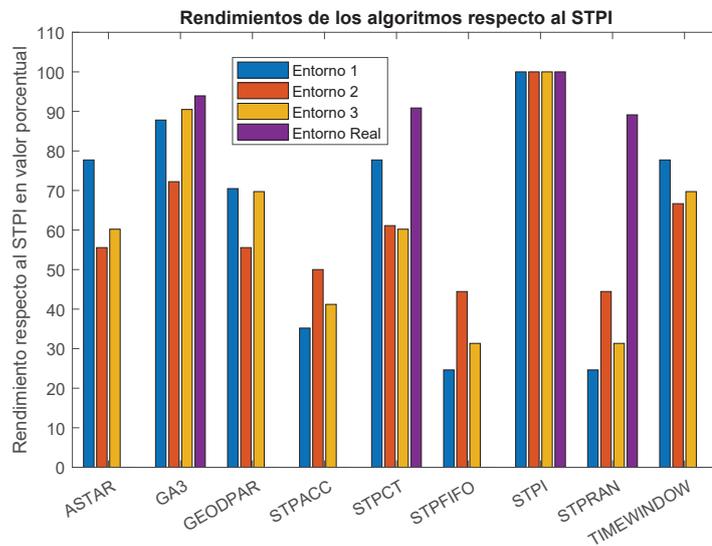


Figura 11.2.10: Visualización del movimiento de los AGV durante la ejecución de las tareas T9\_3-2 y T9\_2-1, utilizando el algoritmo *GA3* en el entorno real

En esta tarea, el algoritmo *STPRAN* y *STPCT* toman el camino más corto para el AGV 1, como se observa en la Figura 11.2.9. Sin embargo, en cambio, como se observa en la Figura 11.2.10, el *GA3* decide que el AGV 1 tome el camino de la izquierda, optimizando el tiempo del AGV 2 a costa de perjudicar el tiempo del AGV 1, logrando que el tiempo medio *TMedV* del conjunto sea menor.

### 11.3 Análisis de los resultados



**Figura 11.3.1:** Rendimiento de todos los algoritmos en los entornos en relación al *STPI*

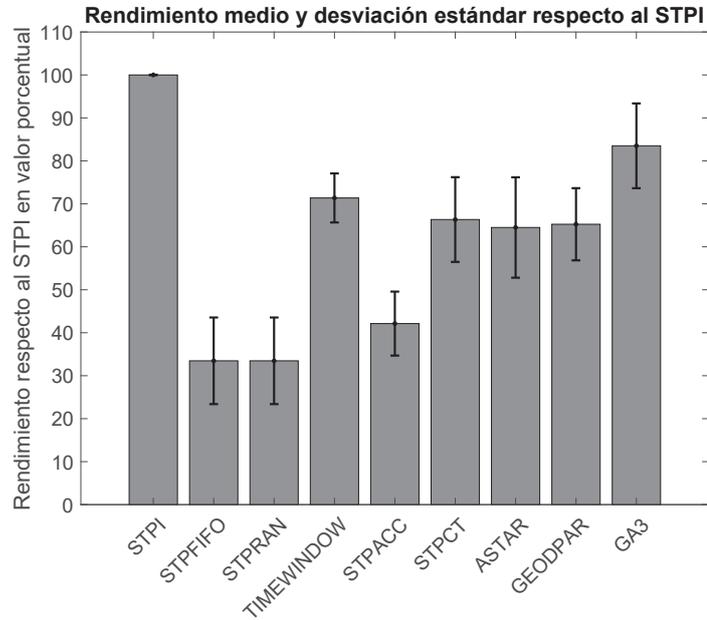
Para calcular el rendimiento de los algoritmos de Gestión de flotas en cada entorno, obteniendo un valor único que pueda ser comparado entre diferentes entornos, al igual que en el apartado 7.4, se utiliza uno de los indicadores específicos de cada entorno y se compara con el rendimiento obtenido por el algoritmo de referencia *STPI*. Se obtiene un valor porcentual del rendimiento en relación al valor máximo teórico. Los indicadores utilizados para cada entorno son los siguientes:

- Entorno 1: Número máximo de misiones completadas, mediante la variable  $N_{MisCo}$ .
- Entorno 2: Número mínimo de AGV necesario para completar las misiones  $N_{AGV}$ .

- Entorno 3: Número máximo de misiones completadas, mediante la variable *NMisCo*.
- Entorno Real: Tiempo de trayecto o viaje medio, mediante la variable *TMedV*.

Como se observa en la Figura 11.3.1, que indica el rendimiento de todos los algoritmos en los entornos en relación al *STPI*, el algoritmo de Gestión de flotas generado por el algoritmo *GA3* es el que ha obtenido un mayor desempeño en todos los entornos en comparación con el algoritmo de rendimiento máximo, el *STPI*. Al analizar cada entorno por separado, podemos detallar las diferencias obtenidas:

- **Entorno 1:** El algoritmo generado por el *GA3* ha logrado un rendimiento del 87,81%. Sin embargo, los algoritmos de referencia que tienen un mejor rendimiento en este entorno (el *TIMWINDOW*, el *STPCT* y el *ASTAR*) han obtenido un rendimiento del 77,72%. Por lo tanto, el algoritmo propuesto, al generar un Gestor de flotas adaptado a este entorno, ha logrado mejorar el rendimiento en un 10,09% en comparación con estos algoritmos que no se adaptan al entorno, lo cual representa una mejora significativa.
- **Entorno 2:** El algoritmo generado por el *GA3* ha obtenido un rendimiento del 72,22%. Sin embargo, el algoritmo de referencia que tiene un mejor rendimiento en este entorno, el *TIMWINDOW*, ha obtenido un 66,67%. Por lo tanto, el algoritmo propuesto, al generar un Gestor de flotas adaptado a este entorno, ha logrado mejorar el rendimiento en un 5,05% en comparación con estos algoritmos que no se adaptan al entorno, lo cual representa una mejora significativa.
- **Entorno 3:** El algoritmo generado por el *GA3* ha obtenido un rendimiento del 90,49%. Sin embargo, los algoritmos de referencia que tienen un mejor rendimiento en este entorno (el *GEOPAR* y el *TIMWINDOW*) han obtenido un rendimiento del 77,72%. Por lo tanto, el algoritmo propuesto, al generar un Gestor de flotas adaptado a este entorno, ha logrado mejorar el rendimiento en un 20,77% en comparación con estos algoritmos que no se adaptan al entorno, lo cual es especialmente significativo en este entorno complejo.
- **Entorno real:** En este entorno real, el algoritmo generado por el *GA3* ha obtenido un rendimiento del 93,94%. Sin embargo, el algoritmo de referencia que tiene un mejor rendimiento en este entorno, el *STPCT*, ha obtenido un 90,87%. Por lo tanto, el algoritmo propuesto, al generar un Gestor de flotas adaptado a este entorno, ha logrado mejorar el rendimiento en un 3,07% en comparación con estos algoritmos que no se adaptan al entorno, lo cual representa una mejora moderada debido a las limitadas posibilidades de mejora debido a la simplicidad del entorno.



**Figura 11.3.2:** Rendimiento medio y desviación estándar de todos los algoritmos en los entornos en relación al *STPI*

Para obtener un único indicador de rendimiento de cada algoritmo, se presentan en la Figura 11.3.2 y en la Tabla 11.3.1 el rendimiento medio en comparación al rendimiento máximo ideal y su desviación estándar. Gracias a estos datos, es posible clasificar los algoritmos según su rendimiento.

Algoritmo	Media	Desviación Estándar
<i>STPFIFO</i>	33,47	10,08
<i>STPRAN</i>	33,47	10,08
<i>STPACC</i>	42,13	7,44
<i>ASTAR</i>	64,50	11,69
<i>GEOPAR</i>	65,25	8,40
<i>STPCT</i>	66,35	9,86
<i>TIMEWINDOW</i>	71,37	5,71
<i>GA3</i>	83,51	9,87
<i>STPI</i>	100,00	0,00

Table 11.3.1: Resultados de todos los algoritmos en los 3 entornos

Finalmente, si observamos el rendimiento medio en todos los algoritmos de la Figura 11.3.2 y en la Tabla 11.3.1, observamos que el rendimiento medio del algoritmo generado por el *GA3* respecto al *STPI* es del 83,51%. El algoritmo de referencia con el mayor rendimiento medio es el *TIMWINDOW*, con un valor del 71,37%. Por lo tanto, en los 3 entornos analizados, el rendimiento del algoritmo generado por el *GA3* es un 12,14% mejor respecto al *STPI*. Se puede constatar que, en los entornos analizados, el algoritmo propuesto que genera Gestores de flotas adaptados al entorno, obtiene un rendimiento superior. Por lo tanto, la adaptación al medio ha otorgado una mayor reducción de costes en los sistemas intralógicos propuestos.

## 11.4 Prueba de hipótesis

Una vez expuestas las evidencias de los resultados del algoritmo generador de Gestores de flotas y de los algoritmos de referencia, se procede a realizar una prueba de hipótesis de la hipótesis de esta tesis (**la generación de algoritmos Gestión de flotas específicamente adaptados para cada entorno permitiría una toma de decisiones específica para el entorno, mejorando así su rendimiento global y reduciendo en consecuencia los costes de la instalación**).

Para realizar la prueba de hipótesis, se utilizarán el rendimiento de los algoritmos adaptativos y no adaptativos utilizando los datos obtenidos de diferentes entornos. En particular, se usarán los datos producidos por el algoritmo propuesto *GA3* para evaluar a los algoritmos que se adaptan al entorno. En contraposición, los resultados de los algoritmos de referencia se utilizarán para evaluar a los algoritmos que no se adaptan al entorno. Es importante destacar que no se tendrán en cuenta los datos de carga computacional, ya que su impacto en la reducción de costos es mínimo en comparación con el rendimiento.

En este apartado, procedemos primeramente a exponer cómo se ha realizado la prueba de hipótesis, posteriormente se realizan los cálculos y conclusiones de la realización de la prueba de hipótesis.

### 11.4.1 Metodología de cálculo la prueba de hipótesis

Para llevar a cabo la prueba de hipótesis, seguimos los siguientes pasos:

1. **Formulación de la hipótesis:** La hipótesis nula ( $H_0$ ) y la hipótesis alternativa ( $H_1$ ) se establecen como:

$$H_0 : \mu_1 - \mu_2 \leq 0 \quad (11.4.1)$$

$$H_1 : \mu_1 - \mu_2 > 0 \quad (11.4.2)$$

Donde  $\mu_1$  es la media de los resultados del algoritmo que se adapta y  $\mu_2$  es la media de los resultados del algoritmo que no se adapta. La hipótesis nula afirma que no hay diferencia o que el algoritmo que no se adapta tiene un desempeño igual o mejor que el que se adapta, mientras que la hipótesis alternativa afirma que el algoritmo que se adapta tiene un mejor rendimiento que el que no se adapta.

2. **Selección de un nivel de significancia:** El nivel de significancia ( $\alpha$ ) se selecciona previamente y se suele establecer en 0,05 o 0,01. Por ejemplo, si seleccionamos un nivel de significancia de 0,05, entonces estamos dispuestos a aceptar un riesgo del 5%.

3. **Recopilación de datos:** Se recopilan los datos de los resultados de ambos algoritmos en una muestra de tamaño  $n$  y se calculan la media muestral ( $\bar{x}_1$  y  $\bar{x}_2$ ) y la desviación estándar muestral ( $s_1$  y  $s_2$ ). Los datos utilizados son los mostrados en la Tabla 11.3.1 del Apartado 11.3. No se consideran los datos de la carga computacional dado que su impacto en la reducción de costos es despreciable respecto a los datos expuestos en la Tabla 11.3.1.

4. **Selección de una prueba estadística:** Se utiliza una prueba  $t$  ya que se desconoce la desviación estándar poblacional. Se utiliza una prueba para muestras independientes, ya que los datos provienen de dos grupos diferentes.

5. **Cálculo de la estadística de prueba:** Se calcula la estadística de prueba utilizando los datos de la muestra y la fórmula correspondiente a la prueba  $t$  para muestras independientes. La estadística de prueba es:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (11.4.3)$$

Donde:

- $t$ : Estadística de prueba.
- $\bar{x}_1$  y  $\bar{x}_2$ : Medias muestrales de los dos grupos, los que se adaptan y los que no.
- $s_1$  y  $s_2$ :  $s_1$  es la desviación estándar del algoritmo que se adapta y  $s_2$  es la desviación estándar de los algoritmos que no se adapta.
- $n_1$  y  $n_2$ :  $n_1$  es el tamaño de la muestra del algoritmo que se adapta y  $n_2$  es el tamaño de la muestra del algoritmo que no se adapta.
- $\Delta$ : Diferencia entre las medias asumida bajo la hipótesis nula y se establece en 0.

6. **Toma de decisión:** Se compara la estadística de prueba con el valor crítico de la distribución  $t$  con  $n_1 + n_2 - 2$  grados de libertad y un nivel de significancia de  $\alpha$ . Si la estadística de prueba es mayor que el valor crítico, entonces se rechaza la hipótesis nula y se acepta la hipótesis alternativa, lo que indica que el algoritmo que se adapta da mejores resultados que el que no se adapta. De lo contrario, no se puede rechazar la hipótesis nula y no se puede afirmar que el algoritmo que se adapta da mejores resultados que el que no se adapta.

Para determinar el valor crítico de la distribución  $t$  para  $\alpha = 0.05$  y  $n_1$  y  $n_2$ , es necesario calcular los grados de libertad de la prueba. Para una prueba  $t$  para muestras independientes con  $n_1$  y  $n_2$  tamaños de muestra y  $\alpha$  nivel de significancia, los grados de libertad se calculan como:

$$df = n_1 + n_2 - 2 \tag{11.4.4}$$

Utilizando una tabla de distribución  $t$  con  $df$  y  $\alpha = 0,05$ , se puede encontrar el valor crítico de la distribución  $t$  correspondiente.

### 11.4.2 Resultados de prueba de hipótesis

Una vez expuesta la metodología de cálculo para la prueba de hipótesis, se procede a realizar la prueba.

Como primer paso, se seleccionan los algoritmos que forman parte del grupo los que no se adaptan al medio  $\mu_2$ . En este caso, se han escogido los algoritmos no convencionales, ya que son los que han obtenido un mayor rendimiento.

Para determinar el valor crítico de la distribución  $t$  con  $\alpha = 0,05$  y  $n_1 = 3$  y  $n_2 = 9$ , es necesario calcular los grados de libertad de la prueba. Para una prueba  $t$  con muestras independientes de tamaño  $n_1$  y  $n_2$  y nivel de significancia  $\alpha$ , se calculan los grados de libertad mediante la ecuación 11.4.4.

En este caso, los grados de libertad serán  $df = n_1 + n_2 - 2 = 10$ .

Utilizando una tabla de distribución  $t$  con  $df = 10$  y  $\alpha = 0,05$ , se puede encontrar el valor crítico de la distribución  $t$  correspondiente. La tabla indica que el valor crítico de la distribución  $t_{crit}$  es aproximadamente 2,228.

Finalmente, se aplica la Ecuación 11.4.3 para realizar el cálculo correspondiente. Como resultado, obtenemos un valor  $t$  de 2,862. Este valor, siendo superior al valor crítico  $t_{crit}$  de 2,228, nos lleva a rechazar la hipótesis nula y a aceptar la hipótesis alternativa. Esta conclusión indica que los algoritmos que se adaptan producen un rendimiento mayor a los que no se adaptan. Por consiguiente, la hipótesis de la tesis se valida mediante esta prueba de hipótesis.

## Capítulo 12

# Conclusiones

### Resumen

En este capítulo se expondrán las conclusiones de esta tesis. Concluyendo que la metodología de generar un Gestor de flotas específico para cada entorno otorga un mayor rendimiento y, por lo tanto, una mayor reducción de costes.

El objetivo principal de esta tesis doctoral es la reducción de costes en el entorno intralogístico, para lograrlo se basa en la hipótesis de que la generación de algoritmos Gestión de flotas específicamente adaptados para cada entorno permitiría una toma de decisiones específica para el entorno, mejorando así su rendimiento global. Cambiando la metodología habitual de generación de un algoritmo pensando en una tipología de entorno, a la generación automática de un Gestor de flotas específicamente adaptado a un entorno específico.

Para lograr el objetivo principal, se realizan diferentes subobjetivos para concluir mediante una prueba de hipótesis que se ha completado el objetivo principal. Como primer subobjetivo, se define el entorno de trabajo de actuación de esta tesis. Especificando el entorno intralogístico de aplicación de la Gestión de flotas

Como segundo subobjetivo, se pretende comprobar la reducción de costes, desarrollando metodologías de medición del rendimiento de un Gestor de flotas en un entorno. Se concluye con el desarrollo de un simulador, capaz de emular el comportamiento de los AGV controlados por un Gestor de flotas en un entorno específico, pudiendo extraer el rendimiento. Además, también es capaz de calcular el rendimiento máximo teórico.

Como tercer subobjetivo, para obtener un desempeño de referencia, se estudia el rendimiento de diferentes Gestores de flotas de referencia, seleccionando los algoritmos de Gestor de flotas, exponiendo los casos de uso y realizando su evaluación mediante el simulador de AGV de esta tesis. Se concluye que el tráfico es el tema donde se puede extraer más rendimiento de un entorno, centrándose en dos grandes puntos: la gestión de intersecciones y la generación de las rutas teniendo en cuenta el tráfico. Se decide que la adaptación al entorno se realizará en estos dos aspectos.

Como cuarto subobjetivo, se proponen los algoritmos de esta tesis mediante la exposición de diferentes algoritmos de complejidad progresiva. El primero se centra en la generación de un gestor de intersecciones adaptado a un entorno simplificado, el segundo se centra en generar un gestor de flotas que solo adapta la gestión de intersecciones en un entorno completo específico, y finalmente se expone el algoritmo *GA3*, que es el algoritmo generador de Gestores de flotas capaz de adaptar sus rutas y la gestión de las intersecciones al entorno requerido.

Como quinto y último subobjetivo, se realiza la evaluación del algoritmo expuesto *GA3* en tres entornos simulados y uno real, obteniendo mejores resultados que los demás Gestores de flotas de referencia en todos los entornos. Finalmente, se valida la hipótesis de esta tesis mediante una prueba de hipótesis,

concluyendo que generar algoritmos de Gestores de flotas adaptados al entorno puede mejorar el rendimiento global, cumpliendo así el objetivo de esta tesis de reducción de costes.

La metodología expuesta en esta tesis se basa en la generación de Gestores de flotas adaptados a un entorno específico, centrándose en la adaptación del tráfico generado en dos aspectos clave: la gestión de las intersecciones y el cálculo de rutas. El objetivo es maximizar el rendimiento del Gestor de flotas de acuerdo con los criterios de evaluación específicos de cada entorno. Además, el Gestor de flotas generado también se adapta al criterio de evaluación de cada entorno.

Para maximizar el rendimiento teniendo en cuenta el tráfico, se incluye en el Gestor de flotas un algoritmo que continuamente extrae la posición futura estimada de cada AGV, teniendo en cuenta su dinámica, la detección perimetral de los sensores de los AGV y las decisiones que tomaría el gestor de intersecciones. Esto permite considerar el tráfico en diferentes partes del algoritmo, como el cálculo de rutas o la estimación del tiempo de llegada. Así, el Gestor de flotas adaptado al entorno puede maximizar el rendimiento de acuerdo con los criterios específicos de cada entorno al tener en cuenta el tráfico.

El primer factor en el que se centra el algoritmo generador de Gestores de flotas adaptados al entorno es la gestión de las intersecciones. Para lograrlo, se utiliza un algoritmo genético que adapta el comportamiento según la instalación mediante la simulación del entorno, considerando los AGV, el *Roadmap* y las tareas. El Gestor de flotas generado se adapta al entorno, teniendo en cuenta sus criterios de evaluación. Durante esta adaptación, se utiliza un generador de rutas que decide las rutas considerando el tráfico generado, utilizando la posición futura estimada y teniendo en cuenta el gestor de intersecciones. En este proceso de adaptación, tanto la gestión de intersecciones como la generación de rutas se adaptan al entorno.

Una vez adaptado el gestor de intersecciones, éste decide el paso de los AGV en cada intersección, considerando la ocupación de la intersección, los destinos de los AGV en otras intersecciones y el tiempo estimado de llegada de cada AGV según su tiempo objetivo. Utilizando estos datos, se selecciona la prioridad de paso de los AGV, maximizando el rendimiento de la flota de AGV de acuerdo con los criterios de evaluación específicos de cada entorno.

El segundo factor en el que se centra el algoritmo generado es el cálculo de rutas. El método utilizado para el cálculo de rutas mediante ventanas de tiempo es utilizado por el Gestor de flotas *TIMEWINDOW* [13]. Este enfoque incor-

pora el tráfico futuro en el cálculo de las rutas, añadiendo tiempo adicional a la ruta en las zonas donde se estima que otro AGV circulara. Sin embargo, este cálculo de posición futura no tiene en cuenta el posible tráfico generado. En el algoritmo propuesto, se utiliza una metodología para extraer la posición futura de los vehículos en cada entorno, que tiene en cuenta el movimiento dinámico de los AGV, las posibles detenciones causadas por el tráfico, así como las detenciones causadas por el gestor de intersecciones. Al utilizar esta metodología, se obtienen dos beneficios. En primer lugar, el cálculo de las rutas tiene en cuenta las posibles detenciones generadas por el tráfico. En segundo lugar, al considerar la decisión del gestor de intersecciones en el proceso de aprendizaje, las rutas se adaptan al entorno.

Por tanto, con el algoritmo propuesto en esta tesis, el *GA3*, se genera un Gestor de flotas adaptado al entorno, adaptando tanto la gestión de intersecciones como el generador de rutas al entorno específico. Como resultado, se obtiene un Gestor de flotas que no solo está adaptado al entorno, sino que también calcula las rutas teniendo en cuenta el tráfico estimado. Además, cuenta con un gestor de intersecciones que considera los criterios específicos del entorno para su desempeño.

Enfocándonos en la metodología utilizada para adaptar el gestor de flotas en esta tesis, se emplea un algoritmo genérico cuyo propósito es determinar qué vehículo tiene prioridad de paso en cada intersección y en cada momento. Así, basándonos en el fundamento propuesto en esta tesis, la metodología de selección de prioridad de paso de los vehículos podría modificarse empleando otros métodos, como las redes neuronales o distintas técnicas de aprendizaje por refuerzo. Esta posibilidad representa una potencial mejora a explorar en futuras investigaciones.

Como conclusión final de esta tesis, podemos afirmar que la metodología de generar un Gestor de flotas específico para cada entorno, maximizando el rendimiento respecto a los criterios de evaluación del entorno, teniendo en cuenta los AGV, el *Roadmap* y las tareas, es capaz de obtener un mejor rendimiento que otros algoritmos de Gestión de flotas que no han sido adaptados a un entorno específico. Esto se traduce en una mayor reducción de costes en el funcionamiento de una flota de AGV en un entorno intralogístico.

# Glossary

- $\mu Bat$  Eficiencia de carga de la batería, contando con todos los elementos que intervienen en el proceso de absorción de energía por parte de un AGV. 54
- $A$  Nodo que determina el destino inicial de una misión. 21, 35, 36, 40, 80, 81, 270, 284, 285
- $A^*$  Algoritmo que sirve para determinar el camino más corto en un grafo. 79, 80, 109, 120
- $AAGV$  Aceleración lineal de un AGV. 48, 188, 211
- $ACenAGV$  Aceleración centrípeta máxima de un AGV. 187
- $AESEg$  Ángulo que tiene dos segmentos seguidos de una ruta. 187
- $AGVEst$  Variables de estado de un AGV. 234, 235
- $AGVNA$  Nodo en el que se encuentra un AGV. 234
- $AGVRuta$  Ruta de un AGV. 234, 235
- $AGV_i DetecRD$  Distancia entre el centro del  $AGV_i$  y el punto más lejano de todas las áreas de detección de todos los sensores de detección del  $AGV_i$ . 202
- $AGV_i Detec_z$  Detector perimetral  $z$  del  $AGV_i$ . 202, 203, 269
- $AGV_i Detec_z P_k$  Punto perimetral con el índice  $k$  de la zona de detección del sensor perimetral  $AGV_i Detec_z$ . 202
- $AMaxRec$  Ángulo que un AGV considera como recta y en el que, por tanto, el vehículo reducirá la velocidad. 187
- $AR$  Ángulo de rotación total que describirá un AGV en una curva. 193
- $ARAGV$  Ángulo de rotación en el que se encuentra un AGV en una curva. 193

- ASTAR* Algoritmo de Gestión de flotas Conflict-free Multi-AGV Routing based on Improved A\* Algorithm. 81, 84, 109, 110, 113, 115, 116, 118, 123, 124, 157, 241, 259, 260
- ASeg* Ángulo del segmento. 192
- ASegFin* Ángulo del segmento de salida de la curva. 187
- ASegIni* Ángulo del segmento de entrada a la curva. 187
- $A_n$  Nodo que determina el destino inicial de una misión de la tarea  $n$ . 35, 36, 37, 38, 41, 42, 284, 285
- $A_{n'}$  Nodo  $A$  de la tarea  $n'$ . 38, 284
- $B$  Nodo que determina el destino final de una misión. 21, 35, 36, 40, 80, 81, 284
- $B_n$  Nodo que determina el destino final de una misión de la tarea  $n$ . 35, 36, 37, 38, 41, 42, 284, 285
- $C$  Nodo que determina el punto de salida único. 35, 37, 42, 284, 285
- $C_e$  Valor del criterio de evaluación utilizando  $e$  para cada  $Ind$ . 135, 141, 277
- $C_kG_z$  *Genes*  $z$  del cruce  $k$  que serán utilizados para determinar el comportamiento de cada intersección. 163
- $C_kNG$  Número de *Genes* del cruce  $k$ . 163, 221, 223
- $C_kNGSegi$  Número de *Genes* del segmento  $i$ , del tipo *SegCruce* del cruce  $k$ . 163, 221, 223
- $C_kPos_i$  Posición de los *Genes*  $G_z$  del segmento  $i$  dentro del cruce  $k$  en el vector de posiciones. 221, 222, 223
- CollisionTable* Tabla que determina las posibles colisiones de los AGV que se encuentran en los segmentos del tipo *SegCruce* del cruce. 147
- Cri1* Criterio de evaluación en el que se maximiza el trabajo que pueden realizar los AGV en un entorno específico, independientemente del número de AGV.. 27, 28, 88, 90, 241
- Cri2* Criterio de evaluación en el cual se minimiza el número de AGV necesarios en un entorno específico con un trabajo concreto que realizar. 27, 29, 88, 93, 241
- Cri3* Criterio de evaluación en el que se maximiza el trabajo que pueden realizar los AGV en un entorno específico con un número determinado de vehículos utilizados. 27, 29, 88, 96, 241

- Cri4* Criterio de evaluación en que se evalúa el desempeño para un número específico de AGV y un trabajo específicos.. 27, 29, 88, 102
- Cruce<sub>j</sub>Nodo<sub>i</sub>* Nodo  $i$  que pertenece al cruce  $j$ . Puede ser de tipo: *NCPreInicio*, *NCInicio*, *NCPreFinal* o *NCFinal*. 234
- DA* Distancia recorrida por un AGV, para acelerar desde la velocidad de la curva anterior hasta la velocidad objetivo del segmento. 188
- DADin* Distancia recorrida en el modo dinámico de aceleración. 191
- DAGV* Desaceleración lineal de un AGV. 48, 188, 211
- DAGV<sub>j</sub>* AGV con el identificador  $j$ , que es el que va a determinar si este activa los sensores perimetrales del *AGV<sub>i</sub>*. 202, 203, 271
- DAGV<sub>j</sub>Estru* Rectángulo que determina la estructura mecánica del AGV *DAGV<sub>j</sub>*. 202, 203, 271
- DAGV<sub>j</sub>EstruP<sub>k</sub>* Punto perimetral con el índice  $k$  de la estructura del AGV representada por *DAGV<sub>j</sub>Estru*. 202
- DAGV<sub>j</sub>EstrucRD* Distancia entre el centro de un *DAGV<sub>j</sub>* y el punto más lejano de su estructura *DAGV<sub>j</sub>Estru*. 202
- DD* Distancia recorrida por un AGV para desacelerar desde la velocidad del segmento a la velocidad de la siguiente curva. 188
- DDDin* Distancia recorrida en el modo dinámico de desaceleración. 191
- DECur* Distancia del segmento entre los dos radios de las curvas del nodo anterior y posterior. 188
- DG* Dirección de giro de una curva. 192, 193
- DMaxSubseg* Distancia máxima que puede tener un subsegmento; idealmente, el tamaño del segmento tendría que ser, aproximadamente, el tamaño medio de todos los AGV. 176, 196
- DN* Distancia restante para llegar al siguiente nodo. 194
- DProxZona* Distancia hasta la próxima zona del segmento. 190
- DR* Distancia al centro de rotación de una curva. 193
- DSeg* Longitud del segmento. 186, 188
- DSeg<sub>s</sub>* Longitud del segmento  $s$ . 48
- DVelCon* Distancia recorrida por un AGV en el modo dinámico de velocidad constante. 191

- $D_m$  Nodo que determina el punto de salida múltiple  $m$ . 35, 37, 38, 285
- $D_{m'}$  Nodo que determina el punto de salida múltiple  $m'$ . 35, 38
- Deadlock* Bloqueo permanente de un conjunto de procesos. En este ámbito, se refiere al bloqueo de dos o más AGV que se bloquean la movilidad entre ellos. 146, 161, 219
- $Detec_{R_1, R_2}$  Determina si el  $R_1$  y  $R_2$  se superponen en un plano bidimensional. 200
- Dijkstra* Algoritmo que sirve para determinar el camino más corto en un grafo. 43, 44, 45, 79, 80, 84, 120, 204, 206
- $DisAGV_{i,j}$  Distancia entre el centro del  $DAGV_j$  y el  $AGV_i$ . 202, 203
- $Dseg_{s,sub}$  Longitud del subsegmento  $sub$  del segmento  $s$ . 210
- $EDesp$  Energía consumida por un AGV desde su última carga de baterías. 54
- $EDespm_m$  Energía consumida por las misiones  $m$ . 56
- $FMut(PMut)$  Función que genera un valor aleatorio para ser añadido a una  $Gen$  con probabilidad de mutación de  $PMut$ . El valor se genera a partir de una función gaussiana. 139
- $FX_j$  Posición del nodo final del segmento  $j$  en el eje  $X$ . 176
- $FY_j$  Posición del nodo final del segmento  $j$  en el eje  $Y$ . 176
- $GA1$  Algoritmo de Gestión de flotas propuesto en la FASE I. xiii, 11, 127, 128, 132, 133, 140, 149, 152, 153, 155, 157, 158, 159, 162, 163, 165, 219, 220, 225, 232, 235
- $GA2$  Algoritmo de Gestión de flotas propuesto en la FASE II. xiii, xiv, 11, 155, 156, 157, 162, 163, 165, 167, 168, 169, 170, 171, 173, 174, 219, 220, 225, 227, 228, 232, 235, 241, 246, 249, 250, 283
- $GA3$  Algoritmo de Gestión de flotas propuesto en la FASE III. xiv, 12, 173, 174, 179, 194, 218, 219, 220, 223, 233, 235, 236, 237, 238, 239, 240, 241, 244, 245, 246, 247, 248, 249, 250, 252, 255, 256, 257, 258, 259, 260, 261, 266, 268
- GEOPAR* Algoritmo de Gestión de flotas Dynamic Geodesic Environment Partition. 81, 83, 109, 110, 113, 116, 118, 123, 124, 126, 157, 168, 169, 178, 241, 246, 259, 260
- $GIni$  *Genes* iniciales, de los que se conoce su funcionamiento. 220, 221
- $GIniSeg$  *Genes* iniciales de un segmento, de los que se conoce su funcionamiento.

- 220, 221
- $GIni_k$  El valor de un *Gen* inicial, del que se conoce su funcionamiento, con índice  $k$ . 221
- $GIni_z$  Valor de un *Gen* inicial, del que se conoce su funcionamiento, con índice  $z$ . 160, 221, 223
- $GMax$  Valor máximo de cada uno de los *Genes*. 134, 135
- $GMin$  Valor mínimo de cada uno de los *Genes*. 134, 135
- $G_z$  El *Gen* con identificador  $z$ . 223, 270
- $Gaussian()$  Función gaussiana, siendo  $\mu$  el valor medio y  $\sigma$  la desviación estándar. 139
- Gen* *Gen* de un *Ind* en un algoritmo genético. 135, 139, 160, 272, 273
- GenCru* *Genes* utilizados para el comportamiento de la gestión del tráfico. 141, 145
- GenCru<sub>i</sub>* *Gen i* utilizado para el comportamiento de la gestión del tráfico. 163
- GenCru<sub>k,c,i</sub>* Datos que determinan el comportamiento con índice  $i$ , del segmento  $c$  del cruce  $k$ . 232
- GenSeg<sub>c,i</sub>* Datos que determinan el comportamiento del cruce, en este caso con el identificador  $i$  del segmento  $c$ . 145
- $Gen_g I_j$  *Ind j* de la Generación  $g$ . 139, 141, 222
- $Gen_g I_j Eval$  Indica la evaluación del *Ind j* de la Generación  $g$ . 135, 136, 139, 141, 223, 224
- $Gen_g I_j G_z$  *Gen z* del *Ind j* de la Generación  $g$ . 135, 222, 223
- $Gen_z$  Generación  $z$  formada por *Ind* de un algoritmo genético. 137, 280
- Genes* *Genes* de un *Ind* en un algoritmo genético. 133, 134, 135, 137, 138, 139, 141, 145, 159, 160, 163, 220, 221, 222, 223, 249, 270, 272, 273, 274, 275
- GrafoMapa* Grafo que determina el *Roadmap* del entorno. Está formado por nodos conectados mediante segmentos direccionales. 43, 44, 273
- GrafoMapaResol* Grafo que debe ser resuelto para encontrar la ruta. Este grafo está basado en el *GrafoMapa*. xi, 44, 45, 46, 47, 204, 205, 206, 207, 210, 214
- IDAGV* ID de un AGV. 212, 234, 235, 283

- Ind* Individuos de un algoritmo genético. 133, 134, 135, 136, 137, 138, 139, 141, 160, 161, 163, 223, 225, 270, 273, 274, 275, 276, 279, 287, 288
- IndSel* *Ind* que han pasado a la fase de selección de un algoritmo genético. 136, 137, 141
- IndSel<sub>y</sub>* El *Ind* con índice *y* que ha pasado a la fase de selección de un algoritmo genético. 139, 224
- InfoC<sub>c,i</sub>* Información del estado de ocupación del cruce de segmento *SegCruce<sub>c</sub>* con el índice *i*. 144
- Informacin<sub>AGV</sub>* Información de un AGV que determina su comportamiento, dimensiones, zonas de detección, velocidades, aceleraciones... 43, 44, 46
- InvRutaP* Todos los destinos que forman una ruta con orden invertido. 215, 216, 217
- InvRutaP<sub>0</sub>* El destino 0 de una ruta con orden invertido, siendo este el destino final de la ruta. 215, 217
- InvRutaP<sub>0</sub>C* Conexión de entrada utilizada en el destino *q* de una ruta con orden invertido, siendo el destino final de la ruta. 215, 217
- InvRutaP<sub>q</sub>* El destino *q* de una ruta con orden invertido. 214, 215
- InvRutaP<sub>q</sub>'C* Conexión de entrada utilizada en el destino *q'* de una ruta con orden invertido. 215
- InvRutaP<sub>q</sub>C* Conexión de entrada utilizada en el destino *q* de una ruta con orden invertido. 214, 215
- InvRutaP<sub>q</sub>UC* Representa la conexión de entrada de *InvRutaP<sub>q</sub>UP* asociada a  $P_{InvRutaP_q}C_{InvRutaP_qC}$ . 215
- InvRutaP<sub>q</sub>UP* Representa el ID del nodo de la conexión de entrada al nodo  $N_{InvRutaP_q}$ . 215, 274
- K<sub>i</sub>* Valor que determina la dispersión inicial de los *Genes*. 135
- MISel* *Ind* mutados en la fase de mutación de un algoritmo genético. 138, 139, 141, 274
- MISel<sub>y</sub>* *Genes* del *Ind* *y* de la *MISel*. 224
- MISel<sub>y</sub>G<sub>z</sub>* Valor de los *Genes* con índice *z* del *Ind* *y* de la *MISel*. 139
- MISel<sub>y</sub>G<sub>z</sub>Mut* Indica el valor añadido que se mutará en cada Generación con índice *z* del *Ind* *y* del conjunto *MISel*. 139
- NAGV* Número de AGV que han sido utilizados en un entorno para cumplir

- las tareas y requisitos de este. 27, 28, 29, 88, 91, 93, 108, 109, 110, 112, 113, 275
- NAGVD* Número de AGV disponibles para ser utilizados en un entorno. 111, 112, 113
- NAGVDisTask* Número de AGV asignados a una tarea. 97
- NAGVSeg<sub>c</sub>* Número de AGV que se encuentran circulando por el segmento del tipo *SegCruce* de un cruce. 148
- NAGVUI* Valor de *NAGV* respecto al valor ideal. Este se obtiene mediante el algoritmo *STPI*. 108, 109
- NCFinal* Nodo en el cual un AGV sale fuera de la zona de influencia del cruce. 142, 143, 144, 151, 167, 235, 271, 281
- NCInicio* Nodo que se encuentra en la entrada física del cruce donde se producen las potenciales colisiones entre los AGV. En este punto se le da, o no, permiso para cruzar. 142, 143, 144, 150, 166, 235, 271, 281
- NCPreFinal* Nodo en el que un AGV se encuentra fuera de la zona de colisión de la intersección, pero en el que una detención del vehículo puede provocar un colapso en la intersección. 142, 143, 144, 150, 167, 235, 271, 281
- NCPreInicio* Nodo en el cual un AGV entra en la zona de influencia del cruce. Este se utilizará para que el algoritmo conozca los destinos y características de los vehículos que tienen intención de entrar al cruce antes de que lleguen a él. 142, 143, 144, 150, 166, 234, 271, 281
- NEval* Número de indicadores de evaluación del entorno. 135
- NG* Número de *Genes* de un *Ind*. 134, 137, 141, 221, 223, 280
- NGIniSeg* Número de *Genes* iniciales de un segmento. 221
- NGen* Número de generaciones de *Ind*. 141, 161, 163, 223
- NInd* Número de *Ind* de un algoritmo genético en una Generación. 134
- NInfo* Número total de variables de información del estado de ocupación del cruce. 145, 164, 231
- NInfoC<sub>Seg</sub>* Número de variables de información utilizadas por cada segmento del tipo *SegCruce* de las demás intersecciones. 226
- NInfoC<sub>k</sub>* Número de variables de información del estado de ocupación del cruce *k*. 163, 164, 231
- NInfoC<sub>k</sub>Seg* Número de variables de información del estado de ocupación por

- cada segmento del tipo *SegCruce* del cruce  $k$ . 163
- $NInfoESeg$  Número de variables de información utilizadas para evaluar el segmento *SegCruce*. 145, 163, 226, 230
- $NInfoOSeg$  Número de variables de información utilizadas por cada *SegCruce* de la propia intersección. 145, 163, 226, 231
- $NInfoPSeg$  Número de variables de información utilizada por cada *SegPosCruce* de todas las intersecciones. 226, 231
- $NInfoP_1$  Número total de información extra utilizada en la propuesta 1. 226
- $NInfoP_2$  Número total de información extra utilizada en la propuesta 2. 226
- $NInfoP_3$  Número total de información extra utilizada en la propuesta 3. 226
- $NInfoP_4$  Número total de información extra utilizada en la propuesta 4. 226
- $NInfoSeg$  Número de informaciones que determinan la ocupación del cruce utilizada por el segmento de tipo *SegCruce<sub>c</sub>* con el índice  $i$ . 141, 144, 145
- $NMisCo$  Número de misiones completadas en un entorno, cumpliendo los requisitos del mismo, en un periodo de tiempo específico. 27, 28, 29, 30, 88, 91, 97, 103, 108, 109, 115, 122, 251, 252, 258, 259
- $NRISel$  Número de *Ind* recombinados en la fase de recombinación de un algoritmo genético. 139
- $NSubseg_j$  Número de subsegmentos del segmento  $j$ . 176, 210
- $NT_{max}$  Número de veces que los AGV han llegado a sus destinos después del  $T_{max}$ . 28, 29, 30, 88, 103, 170, 251, 252
- $NT_{obj}$  Número de veces que los AGV han llegado a sus destinos después del  $T_{obj}$ . 28, 29, 88, 93, 97, 112, 115, 116, 169, 170
- $NVLSub_{s,sub,IDAGV}$  Número de ventanas temporales en las que se tiene que aplicar el  $TExtra_{s,sub,v}$  dependiendo del  $TCir_{s,sub,IDAGV}$ . 212, 213
- $N_{AGV}$  Número de AGV necesarios para cumplir las tareas. 36, 37, 38, 39, 122, 258
- $N_{Final}$  Nodo de destino final de una ruta. 43, 44, 46, 47, 51, 204, 207, 209, 214, 217, 278
- $N_{Inicio}$  Nodo de inicio de una ruta. 43, 44, 46, 47, 51, 204, 214, 215, 217, 278
- $N_{InvRutaP_q}$  Destino por el que se ha llegado, según una ruta invertida, al nodo anterior de destino  $q$  de la ruta. 214, 215, 274

- $Nc$  Número total de segmentos del tipo *SegCruce*. 141, 144
- $Nc_k$  Número total de segmentos del tipo *SegCruce* del cruce  $k$ . 163, 221, 223, 226, 228, 230
- $Nc_{k'}$  Número total de segmentos del tipo *SegCruce* del cruce  $k'$ . 226
- $Nd$  Número de destinos de una tarea. 54
- $Nd_n$  Número de destinos de una tarea  $n$ . 55
- $Nk$  Número total de cruces del entorno. 164, 223, 231
- $Nm$  Número de misiones realizadas desde la última carga de baterías. 56
- $Nm_n$  Número de misiones realizadas desde la última carga de baterías de una tarea  $n$ . 56
- $No_k$  Número total de segmentos del tipo *SegPosCruce* del cruce  $k$ . 228
- $NotNMisCo$  Número de misiones no completadas en un entorno, en un periodo de tiempo específico. 103, 251, 252
- $Np_k$  Número total de segmentos del tipo *SegPreCruce* del cruce  $k$ . 228
- $Np_{k'}$  Número total de segmentos del tipo *SegPreCruce* del cruce  $k'$ . 226
- $Ns$  Número de tareas. 36, 54
- $PBat$  Potencia de carga del cargador, suponiendo que la carga es lineal. (En muchas baterías la potencia de carga es constante entre los rangos de trabajo comunes [entre 20 % y 80 %]). 54
- $PDR_{R1,p,R2}$  Determina si el punto  $p$  del rectángulo  $R1$  se encuentra dentro del rectángulo  $R2$ . 199, 200
- $PE_e$  Variable que pondera el criterio de evaluación del entorno  $C_e$ . 135
- $PMut$  Probabilidad de que una Generación mute. Tiene un valor comprendido entre 0 y 1, en el que el valor 0 significa un 0 % de probabilidad de mutación; y 1, un 100 % de probabilidad de mutación. 139
- $PR$  Posición del punto de rotación de un AGV al girar en la curva. 193
- $PRO$  Orientación del punto de rotación de un AGV al girar en la curva. 193
- $PRX$  Posición X del punto de rotación de un AGV al girar en la curva. 193
- $PRY$  Posición Y del punto de rotación de un AGV al girar en la curva. 194
- $PST$  Porcentaje de carga de trabajo del  $Te$  respecto a tiempo de desplazamiento  $TTTare_{Task}$ . xiv, 97, 114, 115, 116, 169, 170, 246, 249

- $PV$  Periodo de cada una de las ventanas temporales. 212, 214
- $P_{InvRutaP_0C_y}$  Tiempo de llegada a la conexión de entrada  $y$  utilizada para el nodo de destino al que se ha llegado según la ruta del nodo anterior de destino 0 de la ruta con orden invertido. Este será el destino final de la ruta. 215
- $P_{InvRutaP_qC_{InvRutaP_qC}}$  Tiempo óptimo de llegada a la conexión de entrada según el cálculo de rutas para el nodo de destino al que se ha llegado según la ruta del nodo anterior de destino  $q$  de la ruta. 214, 215, 274
- $P_{InvRutaP_qC_y}$  Tiempo de llegada a la conexión de entrada  $y$  utilizada para llegar al nodo anterior de destino  $q$  de la ruta con orden invertido. 214
- $P_{NFinal}$  Tiempo mínimo de llegada al nodo  $N_{Final}$  por todas las conexiones de entrada. 47, 208
- $P_{NInicioC_y}$  Tiempo de llegada al nodo  $N_{Inicio}$  por la conexión de entrada  $y$ . 205
- $P_x$  Tiempo mínimo de llegada al nodo  $x$  por todas las conexiones de entrada. 44, 205, 207, 209
- $P_{x'}$  Tiempo mínimo de llegada al nodo  $x'$  por todas las conexiones de entrada. 47, 208
- $P_{x'}C_{y'}$  Tiempo que se estima que tardará un AGV en llegar al nodo  $x'$  por la conexión de entrada  $y'$ . 46, 47, 207, 208, 284
- $P_{x'}C_{y'}S_{j'}$  Tiempo que se estima que tardará un AGV en llegar al nodo  $x'$  por la conexión de entrada  $y'$  y la conexión de salida  $j'$ . 215, 217
- $P_xC_y$  Tiempo que se estima que tardará un AGV en llegar al nodo  $x$  por la conexión de entrada  $y$ . 44, 46, 205, 207, 209, 210, 217, 284
- $P_xC_yS_j$  Tiempo que se estima que tardará un AGV en llegar al nodo  $x$  por la conexión de entrada  $y$  y la conexión de salida  $j$ . 46, 47, 207, 213, 282, 284
- $P_xC_yS_{j'}$  Tiempo que se estima que tardará un AGV en llegar al nodo  $x$  por la conexión de entrada  $y$  y la conexión de salida  $j'$ . 208
- $Pla$  Places de la Red de Petri que ocupa un AGV al entrar en un segmento o nodo de una Red de Petri. 195, 196
- $Pla_1$  Recursos de ocupación de un segmento o nodo de una Red de Petri. 196
- $Pla_2$  Places con ID 2 de una Red de Petri. 196
- $Pla_3$  Places con ID 3 de una Red de Petri. 196

- Pla<sub>4</sub>* *Places* con ID 4 de una Red de Petri. 196
- Places* Lugares, en una Red de Petri, donde se almacenan los recursos. 195, 196, 278, 279, 280
- PosA* Posición del segmento que separa la zona curva inicial y la zona media del segmento. 186, 187, 188, 189
- PosAGVO* Orientación de un AGV. 192
- PosAGVX* Posición de un AGV en el eje X. 192
- PosAGVY* Posición de un AGV en el eje Y. 192
- PosD* Posición del segmento que separa la zona media del segmento y la zona curva final. 186, 187, 188, 189
- PosNIniX* Posición en el eje X del nodo que inicia el segmento. 192
- PosNIniY* Posición en el eje Y del nodo que inicia el segmento. 192
- PosNexSubSeg* Posición del siguiente Subsegmento en el segmento. 186
- PosSeg* Posición de un AGV en el segmento. 186, 191
- RGMinAGV* Radio mínimo de giro de un AGV. Si es inferior, el AGV rotará. 187
- RGirC* Radio de giro que va a comerter el AGV. 186, 187
- RGirCurAnt* Radio de la curva del nodo anterior. 188
- RGirCurSig* Radio de la curva del nodo siguiente. 188
- RGirNodo* Radio de giro del nodo. 187
- RISel Ind* recombinados en la fase de recombinación de un algoritmo genético. 137, 138, 139, 141
- RIsel<sub>y</sub> Ind* con índice *y* recombinado en la fase de recombinación de un algoritmo genético. 224
- RN* Radio predefinido del nodo en el que se encuentra la curva. 193
- RR* Ratio de rotación de un AGV en una curva. 193, 194
- RR* Ángulo del ratio de rotación de un AGV en una curva. 193, 194
- R<sub>1</sub>* Rectángulo con identificador 1. 199, 200, 272
- R<sub>2</sub>* Rectángulo con identificador 2. 199, 200, 272

- Rand*<sub>0-1</sub>() Función aleatorio que otorga un valor entre 0 y 1. 139
- Rec Places* de la Red de Petri que representan el recurso de ocupación de un segmento o nodo de una Red de Petri. 195, 196
- Rec*<sub>1</sub> Recurso con ID 1 de una Red de Petri. 196
- Rec*<sub>2</sub> Recurso con ID 2 de una Red de Petri. 196
- Rec*<sub>3</sub> Recurso con ID 3 de una Red de Petri. 196
- Rec*<sub>4</sub> Recurso con ID 4 de una Red de Petri. 196
- Rec*<sub>z</sub> Valor que determina de qué padre hereda un hijo el *Gen*<sub>z</sub>. 137
- Reco* Arreglo formado por booleanos de dimensión *NG*, que determina la recombinación de dos *glsInd*. 137
- Roadmap* Mapa de carreteras. En este contexto se refiere a los nodos y segmentos que determinan las carreteras por las que puede circular un vehículo. xii, xiii, 20, 26, 61, 62, 65, 66, 79, 80, 88, 92, 99, 120, 175, 179, 180, 182, 183, 191, 196, 267, 268, 273
- RutaP* Todos los destinos que forman una ruta. 214, 215, 216, 217
- RutaP*<sub>*i*</sub> El destino *i* de una ruta. 214
- STPACC* Algoritmo de Gestión de flotas Shortest Time Path with accumulative input priority crossing. 81, 82, 109, 110, 112, 115, 116, 118, 123, 124, 132, 152, 157, 241, 260
- STPCT* Algoritmo de Gestión de flotas Shortest Time Path with collision table priority crossing. xiv, 81, 82, 83, 109, 110, 113, 115, 116, 118, 120, 123, 124, 125, 126, 132, 152, 153, 157, 169, 241, 250, 252, 255, 256, 257, 258, 259, 260
- STPFIFO* Algoritmo de Gestión de flotas Shortest Time Path with accumulative input priority crossing. 81, 82, 109, 110, 112, 115, 116, 118, 119, 123, 124, 132, 157, 241, 260
- STPI* Algoritmo de Gestión de flotas Shortest Time Path without traffic consideration. xiv, 76, 80, 81, 93, 97, 108, 112, 115, 118, 122, 123, 152, 246, 247, 258, 259, 260, 261, 275, 283
- STPRAN* Algoritmo de Gestión de flotas Shortest time Path with random priority crossing. xiv, 80, 82, 109, 110, 112, 115, 116, 118, 119, 123, 124, 132, 152, 157, 241, 250, 251, 252, 253, 254, 255, 256, 257, 258, 260
- SX*<sub>*j*</sub> Posición del nodo de inicio del segmento *j* en el eje *X*. 176

- $SY_j$  Posición del nodo de inicio del segmento  $j$  en el eje  $Y$ . 176
- $S_j$  Segmento  $j$  de un grafo. 44, 205
- $S_{j,l}$  Subsegmento  $l$  del segmento  $j$  de un grafo. 205
- $SegCruce$  Segmento de una intersección formado por un nodo de entrada  $NCInicio$  y un nodo de salida  $NCPreFinal$  o  $NCFinal$ . Los AGV que se encuentran en este tipo de segmento están cruzando la intersección por la zona potencial de colisión. 143, 144, 145, 146, 147, 148, 150, 163, 166, 220, 221, 223, 225, 226, 228, 229, 230, 231, 234, 270, 275, 276, 277, 281, 286
- $SegCruce_c$  Segmento del tipo  $SegCruce$  con el ID  $c$ . 143, 144, 145, 146, 148, 222, 230, 231, 232, 274, 276, 286
- $SegCruce_{c'}$  Segmento del tipo  $SegCruce$  con el ID  $c'$ . 146, 230
- $SegCruce_{cAGV}$  Segmento del tipo  $SegCruce$  de un cruce que se encuentra en la ruta de un AGV. 148, 150, 151, 166, 167
- $SegCruce_{cAGV}Col$  Número de potenciales colisiones que tendría un AGV al entrar por el segmento  $cAGV$ . 148
- $SegCruce_{cSel}$  Segmento del tipo  $SegCruce$  que tiene prioridad de paso en un cruce. 146, 150, 166
- $SegCruce_{eval}$  Segmento del tipo  $SegCruce$  a evaluar con el ID  $eval$ . 143, 144
- $SegPosCruce$  Segmento de un cruce formado por un nodo de entrada  $NCPreFinal$  y un nodo de salida  $NCFinal$ . Los AGV que se encuentran en este tipo de segmento ya han cruzado la intersección, pero todavía se encuentran en la zona de influencia del cruce. 143, 144, 147, 226, 228, 230, 231, 234, 276, 277, 281, 286
- $SegPosCruce_o$  Segmento del tipo  $SegPosCruce$  con el ID  $o$ . 144
- $SegPosCruce_{oAGV}$  Segmento del tipo  $SegPosCruce$  de un cruce que se encuentra en la ruta de un AGV. 150, 151, 167
- $SegPreCruce$  Segmento de una intersección formado por un nodo de entrada  $NCPreInicio$  y un nodo de salida  $NCInicio$ . Los AGV que se encuentran en este tipo de segmento tienen la intención de entrar en la intersección, pero todavía no se encuentran en la zona potencial de colisión. 143, 144, 146, 147, 226, 228, 229, 230, 234, 277, 281, 282, 286, 287
- $SegPreCruce_p$  Segmento del tipo  $SegPreCruce$  con el ID  $p$ . 143, 144, 230
- $SegPreCruce_{pAGV}$  Segmento del tipo  $SegPreCruce$  de un cruce que se encuentra en la ruta de un AGV. 146, 150, 166

- SegPreCruce<sub>psel</sub>* Segmento del tipo *SegPreCruce* de un cruce seleccionado para dar paso a los AGV. 146, 150, 166
- TA* Tiempo requerido para acelerar un AGV, partiendo de la velocidad objetiva del segmento hasta la velocidad máxima permitida en la siguiente curva. 188, 234, 235
- TADin* Tiempo transcurrido en el modo dinámico de aceleración. 191
- TActual* Tiempo actual. 212
- TBat* Tiempo requerido para la carga de la batería de un AGV. 54
- TBatZona* Tiempo de carga de baterías requerido desde la última carga. 56
- TCF<sub>x,y,j</sub>* Tiempo extra requerido por una curva después del nodo de inicio del segmento. 48, 211
- TCI<sub>x,y,j</sub>* Tiempo extra requerido por una curva antes de llegar al nodo de inicio del segmento. 48, 211
- TC<sub>x,y,j</sub>* Tiempo extra de desplazamiento necesario para tomar la curva asociado a  $P_x C_y S_j$ . 46, 47, 48, 49, 207, 209, 210, 211
- TCir<sub>s,sub,IDAGV</sub>* Tiempo estimado, mediante el cálculo de posición futura, que el AGV *IDAGV* ha tardado en cruzar el subsegmento *susb* del segmento *s*. 212, 213, 276
- TD* Tiempo requerido por un AGV para desacelerar desde la velocidad del segmento a la velocidad de la siguiente curva. 188
- TDDin* Tiempo transcurrido en el modo dinámico de desaceleración. 191
- TDin* Periodo en el que se realizará el cálculo dinámico de los AGV. 184, 190
- TEsp* Tiempo en que un AGV se encuentra esperando una nueva misión. 53, 54
- TEstras, sub, v* Tiempo extra que se estima que tardará un AGV en pasar por el subsegmento *susb* del segmento *s*, en la ventana temporal *v*. 212, 213, 276
- TIMEWINDOW* Algoritmo de Gestión de flotas Dynamic Routing of Automated Guided Vehicles in Real-time. 81, 84, 109, 110, 113, 116, 118, 123, 124, 126, 157, 168, 173, 178, 203, 241, 245, 246, 259, 260, 261, 267
- TIni* Tiempo de inicialización de la carga. Los sistemas de carga de baterías requieren un tiempo de inicialización para comenzar a cargar las baterías. 54

- $TLAGV_{s,0}$  Tiempo absoluto de entrada de un AGV al segmento  $s$ . 213, 214
- $TLAGV_{s,sub}$  Tiempo absoluto de entrada de un AGV al subsegmento  $sub$  del segmento  $s$ . 213, 214
- $TL_{s',sub',IDAGV}$  Tiempo absoluto de llegada previsto al subsegmento siguiente del  $sub$  del segmento  $s$  para el AGV con ID  $IDAGV$ . 212
- $TL_{s,sub,idagv}$  Tiempo previsto absoluto de llegada al subsegmento  $sub$  del segmento  $s$  para el AGV con ID  $IDAGV$ . 212
- $TMaxT_{max}$  Tiempo máximo de llegada de los AGV a sus diferentes destinos respecto a sus  $Tmax$  respectivos. Medido en segundos. 30, 103, 251
- $TMaxV$  Tiempo máximo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos. 30, 103, 252
- $TMedT_{max}$  Tiempo medio de llegada de los AGV a sus diferentes destinos respecto a sus  $Tmax$  respectivos. Medido en segundos. 28, 30, 103, 251
- $TMedT_{obj}$  Tiempo medio de llegada de los AGV a sus destinos respecto a sus  $Tobj$  respectivos. Medido en segundos. 28, 29, 88, 93, 97, 111, 112, 113, 114, 115, 116, 169, 283
- $TMedT_{objI}$  Valor de  $TMedT_{obj}$  respecto al obtenido por el algoritmo  $STPI$ . Medido en segundos. 112
- $TMedT_{objRI}$  Valor de  $TMedT_{obj}$  respecto al obtenido por el algoritmo  $STPI$ . Medido en valor porcentual con respecto al algoritmo  $STPI$ . 115, 116
- $TMedT_{objRI}$  Valor de  $TMedT_{obj}$  respecto al obtenido por el algoritmo  $GA2$ . Medido en valor porcentual con respecto al algoritmo  $GA2$ . 169
- $TMedV$  Tiempo medio de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos. 27, 28, 29, 30, 88, 91, 103, 108, 109, 110, 251, 252, 258, 259
- $TMinT_{max}$  Tiempo mínimo de llegada de los AGV a sus diferentes destinos respecto a sus  $Tmax$  respectivos. Medido en segundos. 30, 103, 251
- $TMinV$  Tiempo mínimo de desplazamiento de todos los AGV entre sus diferentes destinos. Medido en segundos. 30, 103, 252
- $TPro_d$  Tiempo que un AGV tiene que permanecer en el destino  $d$  de una tarea, ya sea por el tiempo requerido por el vehículo para realizar una tarea, o por los requerimientos de la instalación. 53
- $TPro_{d_n}$  Tiempo que un AGV tiene que permanecer en el destino  $d$  de la tarea  $n$ , ya sea por el tiempo requerido por el vehículo para hacer una tarea, o por los requerimientos de la instalación. 55

- $TPro_{desp}$  Tiempo de operación que tarda un AGV en realizar una tarea en el lugar donde le espera una nueva misión. 53
- $TS_{x,y,j}$  Tiempo de desplazamiento a la velocidad objetivo sin tráfico del segmento asociado a  $P_xC_yS_j$ . 46, 47, 207, 209, 210
- $TSinTra_{Task, Dest}$  Tiempo de desplazamiento desde el destino de inicio de la tarea  $Task$  hasta el destino  $Dest$  de esa misma tarea, sin considerar el tráfico. 93, 97
- $TTBatZona$  Tiempo de desplazamiento a la zona de carga de baterías. 56
- $TTBatZona_n$  Tiempo de desplazamiento a la zona de carga de baterías requerido al realizar la tarea  $n$ . 56
- $TTM$  Tiempo de desplazamiento un AGV entre todos los destinos de la misión. 53
- $TTTare_{Task}$  Tiempo de desplazamiento total necesario para completar una tarea, sin considerar el tráfico. 97, 114, 115, 277
- $TT_{x,y,j}$  Tiempo extra requerido por el tráfico para recorrer el segmento asociado a  $P_xC_yS_j$ . 207, 209, 210, 213, 214
- $TV_{x,y,j}$  Tiempo de desplazamiento para recorrer desde  $P_xC_y$  hasta  $P_{x'}C_{y'}$ , asociado a la conexión de salida  $P_xC_yS_j$ . 46, 47, 207, 210
- $TVelCon$  Tiempo transcurrido en el modo de velocidad constante dinámica de un AGV. 191
- $T_2$  Transición con ID 2 de una Red de Petri. 196
- $T_{AB}$  Tiempo que tarda un AGV en circular del punto  $A$  al punto  $B$ . 36, 40, 43
- $T_{AB_n}$  Tiempo que tarda un AGV en circular del destino  $A_n$  al destino  $B_n$  de la tarea  $n$ . 36, 41, 42
- $T_{BA}$  Tiempo que tarda un AGV en circular del punto  $B$  al punto  $A$ . 36, 40, 43
- $T_{BA_n}$  Tiempo que tarda un AGV en circular del punto  $B_n$  al punto  $A_n$  de la tarea  $n$ . 36, 41
- $T_{BC}$  Tiempo que tarda un AGV en circular del punto  $B$  al punto  $C$ . 43
- $T_{B_nA_{n'}}$  Tiempo que tarda un AGV en circular del destino  $B_n$  al destino  $A_{n'}$ , donde  $A_{n'}$  es el destino de recogida de la siguiente tarea, que es más óptima en cuanto al rendimiento de la Gestión de flotas. 38
- $T_{B_nC}$  Tiempo que tarda un AGV en circular del punto  $B_n$  de la tarea  $n$  al punto  $C$ . 37, 42

$T_{B_n D_{m'}}$  Tiempo que tarda un AGV en circular del  $B_n$  de la tarea  $n$  al punto múltiple  $D_{m'}$  con ID  $m'$ . 38

$T_{CA}$  Tiempo que tarda un AGV en circular del punto  $C$  al punto  $A$ . 43

$T_{CA_n}$  Tiempo que tarda un AGV en circular del punto  $C$  al punto  $A_n$  de la tarea  $n$ . 37, 42

$T_{D_m A_n}$  Tiempo que tarda un AGV en circular del  $D_m$  del punto de salida múltiple  $m$  al punto  $A_n$  de la tarea  $n$ . 38

$Tdmax$  Desviación del tiempo de llegada a un destino respecto a  $Tmax$ . 218, 229, 230, 234, 236

$Te$  Periodo entre peticiones de una tarea. Medido en segundos. 35, 36, 40, 43, 91, 93, 97, 107, 108, 114, 115, 170, 246, 277

$Te_n$  Periodo entre peticiones de la tarea  $n$ . Medido en segundos. 36, 39, 41, 42, 43, 90, 97

$Temp_{P_{x'}C_{y'}}$  Variable temporal que determina el tiempo de llegada al nodo  $x'$  por la conexión de entrada  $y'$ . 46, 47, 207, 208

$TiempoMuestreo$  Periodicidad del cálculo dinámico de los AGV. 180, 182

$Tmax$  Tiempo máximo de llegada impuesto por el entorno para un destino de una tarea. 25, 28, 29, 30, 97, 98, 102, 103, 115, 169, 170, 218, 229, 249, 251, 252, 276, 283, 285

$Tmax_{Task, Dest}$  Tiempo máximo de llegada impuesto por el entorno para el destino  $Dest$  de la tarea  $Task$ . 97

$Tobj$  Tiempo de llegada objetivo impuesto por el entorno para un destino de una tarea. 24, 28, 29, 93, 97, 111, 112, 115, 170, 246, 249, 276, 283

$Tobj_{Task, Dest}$  Tiempo objetivo de llegada impuesto por el entorno para el destino  $Dest$  de la tarea  $Task$ . 93, 97

$VAGV$  Velocidad actual del AGV. 191

$VCR$  Velocidad máxima a la que puede circular un AGV antes de tener que rotar. 187

$VLSub_{s, sub, IDAGV}$  Indicación de la ventana temporal en la que se estima que un AGV  $IDAGV$  entrará en el subsegmento  $sub$  del segmento  $s$ . 212, 213

$VMC$  Velocidad objetivo de giro en la curva asociada a un nodo. 186, 187

$VMCFin$  Velocidad objetivo de giro en la curva asociada al nodo de llegada

- del segmento. 186, 187
- VMCG* Velocidad máxima de giro que puede describir un AGV con el radio de giro indicado. 187
- VMCIni* Velocidad objetivo de giro en la curva asociada al nodo de partida del segmento. 186, 187
- VMaxAGV* Velocidad máxima de un AGV. 187
- VNodo* Velocidad objetiva del nodo. 187
- VObj* Velocidad objetiva del segmento. Esta es la mínima entre la velocidad máxima de un AGV y la velocidad objetiva del nodo anterior al segmento. 48, 49, 189, 210, 211, 286
- VObjSeg* Velocidad objetiva del segmento. Esta es la mínima entre la velocidad máxima de un AGV y la velocidad objetiva del nodo anterior al segmento. 186, 188
- VObj<sub>-1</sub>* *VObj* de segmento anterior. 48, 211
- VObj<sub>0</sub>* *VObj* del segmento actual. 49, 211
- VSSub<sub>s,sub,IDADGV</sub>* Indicación de la ventana temporal en la que se estima que un AGV *IDAGV* saldrá del subsegmento *sub* del segmento *s*. 213
- VSeg<sub>c</sub>* Valor de prioridad del segmento tipo *SegCruce<sub>c</sub>* con índice *c* de un cruce. 145, 146, 232
- VelCur* Velocidad objetiva de la curva. 48, 211
- cAGV* ID del segmento del tipo *SegCruce* que se encuentra en la ruta de un AGV de un cruce. 148, 234, 235, 281
- cSel* ID del segmento del tipo *SegCruce* que tiene prioridad de paso en un cruce. 146, 234, 235
- inSeg* Segmento del tipo *SegCruce* en el que un AGV pretende entrar en la zona de colisión del cruce. 147, 148
- n* ID de la tarea a realizar. 35, 36, 37, 38, 41, 42, 43, 55, 56, 270, 277, 283, 284, 285
- oAGV* ID del segmento del tipo *SegPosCruce* que se encuentra en la ruta de un AGV. 234, 235
- outSeg* Segmento del tipo *SegCruce* donde puede haber un AGV. 147, 148
- pAGV* ID del segmento del cruce del tipo *SegPreCruce* que se encuentra en la ruta de un AGV. 150, 166, 234, 235

*pSel* ID del segmento del tipo *SegPreCruce* que tiene prioridad de paso. 234, 235

*v(sub)* Ventana temporal que hay que utilizar para el cálculo del tiempo extra. 213

**AGV** Automatic Guided Vehicle. viii, xi, xii, xiii, xiv, 3, 7, 10, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 46, 48, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 70, 71, 72, 73, 74, 75, 81, 82, 83, 84, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 108, 109, 110, 111, 112, 113, 114, 115, 118, 122, 125, 126, 127, 129, 130, 131, 132, 134, 142, 143, 144, 145, 146, 148, 150, 151, 152, 153, 155, 157, 158, 166, 167, 168, 170, 171, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 201, 202, 203, 204, 205, 207, 209, 210, 211, 212, 213, 214, 217, 218, 219, 225, 226, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 242, 243, 244, 245, 246, 247, 248, 251, 252, 254, 255, 256, 257, 258, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 281, 282, 283, 284, 285, 286, 288

**AMR** Autonomous Mobile Robot. 20

**Generación** Ciclo de evolución de los *Ind* de una Población. 133, 135, 136, 139, 141, 273, 274, 275, 277, 287, 288

**Generaciones** Conjunto formado por cada Generación. 133

**Gestión de flotas** Administración de un grupo de vehículos de transporte que han sido reunidos bajo un mismo propósito por una organización. i, ii, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 16, 17, 38, 59, 77, 79, 80, 81, 84, 106, 108, 122, 125, 129, 131, 154, 155, 171, 240, 241, 250, 258, 259, 261, 266, 268, 270, 272, 280, 282, 284

**Gestor de flotas** Algoritmo que se encarga de la administración de un grupo de vehículos de transporte que han sido reunidos bajo un mismo propósito por una organización. Sus funciones abarcan distintas Tareas, como ocuparse del mantenimiento de las unidades de la flota, administrar los sistemas de seguimiento, planificar los recorridos o llevar un control riguroso de los costes asociados a cada decisión en la gestión, entre otras. A través de estas acciones, la gestión de flotas rentabiliza la inversión en vehículos y equipamiento para logística. Su objetivo fundamental es optimizar la eficiencia y productividad de una flota en todas las etapas de su operación. xi, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 24, 25, 26, 27, 34, 38, 60, 61, 62, 66, 67, 79, 84, 86, 90, 112, 118, 127, 128, 132, 141, 146, 150, 155, 157, 163, 166, 174, 178, 179, 180, 182, 189, 217, 218, 236, 237, 248, 259, 265, 266, 267, 268, 288

**Gestores de flotas** Conjunto de Gestor de flotas. ii, 5, 6, 7, 10, 11, 12, 18, 61, 80, 82, 132, 167, 173, 174, 261, 266, 267

**Misión** Consiste en una Tarea concreta que se debe realizar en un tiempo especificado y que sera realizada por un AGV. 22, 23, 24, 288

**Misiones** Conjunto formado por más de una Misión. xi, 22, 23

**Petición** Solicitud para realizar una Tarea, que a su vez genera una Misión. 22

**Población** Conjunto de *Ind* que forman una Generación de un algoritmo genético. 287

**Red de Petri** Consiste en la representación matemática o gráfica de un sistema mediante el uso de eventos discretos en el cual se puede describir la topología de un sistema distribuido. xiii, 194, 195, 196, 278, 279, 280, 284

**Ruta** Conjunto de nodos que determinan el camino que tomará un AGV para ir de un destino a otro. 217

**Subsegmento** Subparte de un segmento. 176, 182, 184, 186, 189, 197, 209, 212, 213, 214, 279, 281

**Subsegmentos** Conjunto de subpartes que forman un segmento. 175, 176, 179, 196, 197, 204, 205, 209, 212, 214, 237

**Tarea** Trabajo concreto que se realiza en una instalación por los AGV. Un ejemplo sería mover una carga de un punto determinado *A* al punto determinado *B*. xi, 21, 22, 23, 24, 288

**Tareas** Conjunto de trabajos que se realizan en una instalación. 21, 22, 287

# Bibliography

- [1] Abichandani, Pramod, Hande Y. Benson, and Moshe Kam: *Decentralized multi-vehicle path coordination under communication constraints*. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.
- [2] Broadbent, BAJ., CB. Besant, SK. Premi, and SP. Walker: *Free ranging agv systems: Promises, problems and pathways*. Proceedings of the 2nd International Conference on Automated Materials Handling, UK: IFS Publication Ltd, 1985.
- [3] Davis, L.D.: *Handbook of genetic algorithms*. Artificial Intelligence, Volume 100, 1998, Pages 325-330, 1997.
- [4] Desrosiers, J. and M. Solomon: *Time window constrained routing and scheduling problems*. Transportation Science, Vol. 22, pp. 1-13, 1998.
- [5] Desrosiers, M. and et al.: *Methods for routing with time windows*. European Journal of Operations Research, Vol. 23, 1986.
- [6] Desrosiers, M. and F. Soumis: *A generalized permanent labelling algorithm for the shortest path problem with time windows*. INFOR, Vol. 26, No. 3, 1988.
- [7] Digani, V., L. Sabattini, C. Secchi, and C. Fantuzzi: *Hierarchical traffic control for partially decentralized coordination of multi-agv systems in industrial environments*. IEEE International Conference on Robotics and Automation (ICRA), 2014.
- [8] Digani, Valerio, Lorenzo Sabattini, and Cristian Secchi: *A probabilistic eulerian traffic model for the coordination of multiple agvs in automatic warehouses*. IEEE Robotics and Automation Letters, Vol. 1, No. 1, January 2016, 2016.
- [9] Digani, Valerio, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi: *Ensemble coordination approach in multi-agv systems applied to industrial*

- warehouses*. Transactions on Automation Science and Engineering, Vol. 12, No. 3, July 2015, 2015.
- [10] Dijkstra, E. W.: *A note on two problems in connexion with graphs*. Numerische mathematik, 1(1):269–271, 1959.
- [11] Fantì, M., A. Mangini, G. Pedroncelli, and W. Ukovich: *Decentralized deadlock-free control for agv systems*. American Control Conference (ACC), 2015. 2015.
- [12] Fogel, David B.: *An introduction to simulated evolutionary optimization*. IEEE Transactions on Neural Networks, Volume 5, Issue 1, 1994, Pages 3-14, 1994.
- [13] Gawrilow, Ewgenij, Ekkehard Köhler, Rolf H. Möhring, and Björn Stenzel: *Dynamic routing of automated guided vehicles in real-time*. Technische Universität Berlin, 2007.
- [14] Goldberg, David E: *Genetic algorithms*. Computer science and operations research, 10:26–32, 1991.
- [15] HE, LN., PH. LOU, XM. QIAN, and R. LIU: *Conflict-free automated guided vehicles routing based on time window*. Computer Integrated Manufacturing Systems, Beijing, 2010.
- [16] Holland, John H: *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [17] Hoshino, S. and H. Seki: *Multi-robot coordination for jams in congested systems*. Robotics and Autonomous Systems, 2013.
- [18] Hui, N.: *Coordinated motion planning of multiple mobile robots using potential field method*. International Conference on Industrial Electronics, Control Robotics (IECR), 2010.
- [19] ISO: *Industrial trucks - safety requirements and verification Part 4: Driverless industrial trucks and their systems*. Standard ISO 3691-4:2020, International Organization for Standardization, Geneva, CH, 2020. <https://www.une.org/encuentra-tu-norma/busca-tu-norma/proyecto?c=P0049998>.
- [20] JIA, Fang, Chenglong REN, Yi CHEN, and Zhixiang XU: *A system control strategy of a conflict-free multi-agv routing based on improved a\* algorithm*. Department of Mechanical Engineering, Southeast University Nanjing, 2017.
- [21] Kavraki, L., P. Svestka, J. C. Latombe, and M. Overmars: *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566–580,

- 1996.
- [22] Kennedy, James and Russell Eberhart: *Particle swarm optimization*. Encyclopedia of machine learning, pages 760–766, 2010.
  - [23] Koza, John R: *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992.
  - [24] Krishnakumar, K. and David E. Goldberg: *Control system optimization using genetic algorithms*. Journal of Guidance, Control, and Dynamics, Volume 15, Issue 3, 1992, Pages 735-740, 1992.
  - [25] LaValle, Steven M. and Seth A. Hutchinson: *Optimal motion planning for multiple robots having independent goals*. IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, 1998.
  - [26] LI, WG. and X. SU: *Agv path planning based on improved a\* algorithm*. Modern Manufacturing Engineering, Vol. 29, pp. 33-36, 2015.
  - [27] Li, X. and F. Lara-Rosano: *Adaptive fuzzy petri nets for dynamic knowledge representation and inference*. Expert Systems with Applications, Volume 19, Issue 3, 2000, Pages 235-241, 2000.
  - [28] Li, Xiaou, Wen Yu, and Felipe Lara-Rosano: *Dynamic knowledge inference and learning under adaptive fuzzy petri net framework*. IEEE Transactions on Systems, Man, and Cybernetics - Part C - Applications and Reviews, Volume 30, Issue 4, 2000, Pages 442-450, 2000.
  - [29] Manca, S., A. Fagiolini, and L. Pallottino: *Decentralized coordination system for multiple agvs in a structured environment*. 18th World Congress of the International Federation of Automatic Control (IFAC 2011), 2011.
  - [30] Masoud, Ahmad A.: *Decentralized self-organizing potential field-based control for individually motivated mobile agents in a cluttered environment: A vector-harmonic potential field approach*. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 37, No. 3, 2007.
  - [31] McCall, John: *Genetic algorithms for modelling and optimisation*. Journal of Computational and Applied Mathematics, Volume 184, Issue 1, 1 December 2005, Pages 205-222, 2005.
  - [32] Mitchell, Melanie: *Applications of genetic algorithms*. AI expert, 8(6):24–27, 1993.
  - [33] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.: *Deep reinforcement learning*. Nature, 518(7540):529–533, 2015.

- [34] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, *et al.*: *Human-level control through deep reinforcement learning*. *Nature*, 518(7540):529–533, 2015.
- [35] Puterman, Martin L: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley Sons, 2014.
- [36] Rechenberg, Ingo: *Evolutionstrategie—optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog Stuttgart, 15, 1973.
- [37] Sabattini, Lorenzo, Valerio Digani, Cristian Secchi, and Cesare Fantuzzi: *Hierarchical coordination strategy for multi-agv systems based on dynamic geodesic environment partitioning*. International Conference on Intelligent Robots and Systems (IROS), 2016.
- [38] Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov: *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347, 2017.
- [39] Secchi, C., R. Olmi, F. Rocchi, and C. Fantuzzi: *Dynamic routing strategy for the traffic control of agvs in automatic warehouses*. IEEE International Conference on Robotics and Automation (ICRA), 2015.
- [40] SUN, Q., H. LIU, and Q. YANG: *On the design for agvs: Modeling, path planning and localization*. IEEE International Conference on Mechatronics and Automation, China, pp. 163-169, 2011.
- [41] Sutton, Richard S and Andrew G Barto: *Reinforcement learning*. Handbook of learning and approximate dynamic programming, 1:1–8, 2004.
- [42] Watkins, Christopher JCH and Peter Dayan: *Q-learning*. *Machine learning*, 8(3-4):279–292, 1992.
- [43] Whitley, Darrell: *Genetic algorithms: Reviews, implementations and applications*. *Statistics and Computing*, Volume 4, Issue 2, 1994, Pages 65-85, 1994.
- [44] Yang, P., R. Freeman, and K. Lynch: *Multi-agent coordination by decentralized estimation and control*. *IEEE Transactions on Automatic Control*, Vol. 53, No. 11, pp. 2480–2496, 2008.
- [45] Zheng, K., D. Tang, W. Gu, and M. Dai: *Distributed control of multi-agv system based on regional control model*. *Production Engineering*, pp. 1–9, 2013.