

UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

FINAL PROJECT

**Automatic Guided Vehicle System for
warehouse management:
implementation and simulation**

Authors:

Andrea Spreafico - 793317 - a.spreafico13@campus.unimib.it

Francesco Prete - 793389 - f.prete4@campus.unimib.it



Contents

1	Introduzione al problema	4
1.1	Panoramica	4
1.2	Obiettivi	4
1.3	Strumenti utilizzati	4
2	Stato dell'arte	5
2.1	Agente	5
2.2	Sistema	6
2.3	Sistema complesso	7
2.4	Simulazione sistema	7
2.5	Sistema multiagente	8
2.5.1	Modelli di interazione tra agenti	9
2.5.2	Tipologie di ambienti	10
2.6	Automated Guided Vehicle	11
2.7	Automated Guided Vehicle System	12
2.7.1	Classificazione di un AGVS	12
2.7.2	Performance di un AGVS	12
2.8	Lavori precedenti	13
3	Descrizione del dominio	14
3.1	Azienda	14
3.2	Tipologia di lavoro e di ordini	14
3.3	Ambiente	14
3.4	Lista degli ordini	16
4	Descrizione del modello	17
4.1	Ambiente del simulatore	17
4.2	Modellazione agenti	20
4.2.1	Stati assumibili	21
4.2.2	Transizioni tra stati	23
4.3	Gestione navigazione	24
4.3.1	Esempio navigazione - Libero	25
4.3.2	Esempio navigazione - Ostacolo	26
4.4	Gestione conflitti	27
4.4.1	Esempio conflitto - Cambio direzione	28

4.4.2	Esempio conflitto - Attesa	29
4.5	Gestione area scarico merci	30
4.6	Gestione area di attesa	30
4.7	Gestione ordini e agenti	31
4.7.1	Behaviour type 1	31
4.7.2	Behaviour type 2	31
4.7.3	Behaviour type 3	31
5	Implementazione simulatore	32
5.1	Interfaccia simulatore	32
6	Simulazione	34
6.1	Configurazioni simulate	34
6.2	Lista degli ordini per simulazione	35
6.3	Metriche di valutazione	37
7	Analisi dei risultati	38
7.1	Analisi generale delle configurazioni	39
7.1.1	Timestep per ciascuna configurazione	39
7.1.2	Conflitti per ciascuna configurazione	40
7.2	Analisi singoli AGV delle configurazioni	41
7.2.1	Articoli elaborati da ciascun AGV	41
7.2.2	Timestep di attesa per ciascun AGV	43
8	Conclusioni	45
8.1	Lavori e sviluppi futuri	46

List of Figures

1	Interazione di un agente con l'ambiente esterno	5
2	Ciclo di vita di una simulazione	7
3	Interazione di più agenti con l'ambiente	8
4	Modelli di interazione tra agenti	9
5	Diversi modelli di AGV	11
6	Mappa fisica del magazzino	15
7	Numero di ordini per ciascun cliente	16
8	Rappresentazione della mappa del magazzino sul simulatore .	17
9	Transizioni tra stati di un agente	23
10	BFS: Ordine in cui i nodi vengono esplorati	24
11	Area scarico merci: MI, FI, CO	30
12	GUI del simulatore implementato	32
13	GUI del simulatore durante esecuzione	33
14	Numero di ordini associati ad ogni cliente	35
15	Numero di diversi articoli per cliente	36
16	Tempo necessario al completamento degli ordini per ciascuna configurazione	39
17	Rapporto tra tempo richiesto e conflitti avvenuti in ciascuna configurazione	40
18	Numero di articoli elaborati da ciascun AGV con BT1	41
19	Numero di articoli elaborati da ciascun AGV con BT2	42
20	Numero di articoli elaborati da ciascun AGV con BT3	42
21	Numero di step temporali passati in attesa da ciascun AGV per i tre diversi BT	43
22	Numero totale di step passati in attesa dagli AGV di una configurazione	44

List of Tables

1	Diverse tipologie di ambiente	10
2	Diverse tipologie di tecnologie utilizzate per realizzazione AGV	11

1 Introduzione al problema

1.1 Panoramica

IoT, guida autonoma, domotica e tutto quello che riguarda l'automazione di processi (produttivi e domestici) attraverso l'uso di "macchine intelligenti" sono argomenti sempre più presenti nella vita di tutti i giorni e sono destinati, in un futuro molto vicino, ad essere la normalità.

Questo è l'ambito in cui si sviluppa il nostro progetto: l'automazione dei processi produttivi di un'azienda di trasporti e logistica, la LDE S.r.l. di Brembate. Questa società ci ha fornito i dati riguardanti la gestione tecnica, gli ordini e il magazzino e, quindi, tutti i dati utilizzati per lo sviluppo di questo lavoro sono interamente reali e derivati dalla sopracitata azienda. Ad oggi, nella LDE lo smistamento dei pacchi è eseguito in maniera totalmente manuale dai dipendenti, che prevede un grande sforzo lavorativo ed organizzativo.

1.2 Obiettivi

Lo scopo di questo lavoro è quello di implementare, simulare e valutare la realizzazione di un sistema a veicoli a guida autonoma per la gestione semi automatica di un magazzino. In particolare, lo scopo finale del lavoro sarà quello di stimare il numero ottimale di veicoli e il comportamento che questi ultimi dovranno adottare per minimizzare i costi e i tempi di esecuzione nello smistamento degli ordini.

1.3 Strumenti utilizzati

Gli strumenti principali utilizzati per la progettazione e l'implementazione di questo lavoro sono stati:

- Python 3.6.7 [2]
- PycxSimulator [11]
- Tableau [1]
- Github [3]

2 Stato dell'arte

2.1 Agente

L'agente è una qualunque entità in grado di percepire l'ambiente che lo circonda attraverso dei sensori e di eseguire delle azioni attraverso degli attuatori. Ad esempio, in un essere umano alcuni sensori sono gli occhi e le orecchie. Gli attuatori possono essere le mani, i piedi o più in generale i muscoli.

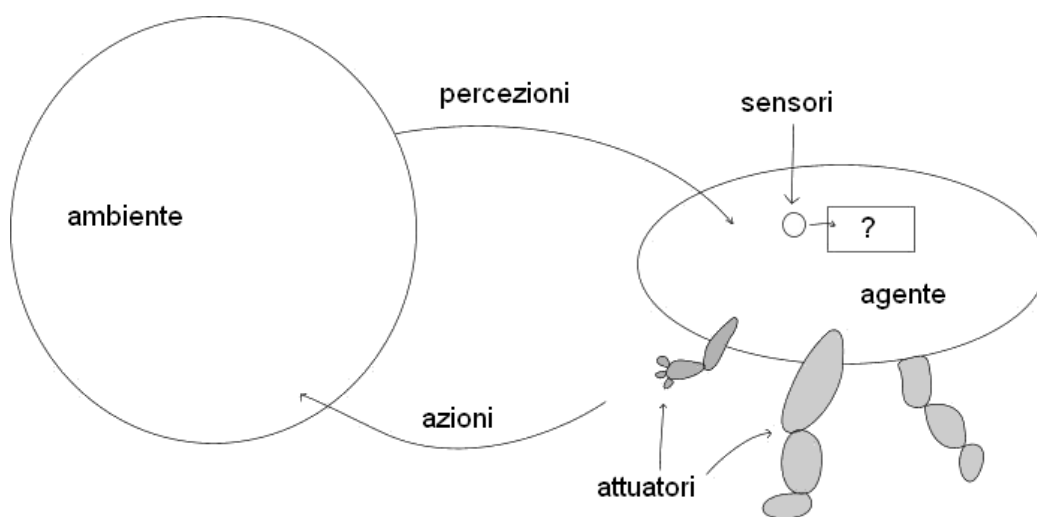


Figure 1: Interazione di un agente con l'ambiente esterno

Nel campo dell'Intelligenza artificiale un obiettivo fondamentale è la realizzazione degli agenti intelligenti (o agenti razionali). Nella fattispecie un agente si definisce intelligente se fa la cosa giusta al momento giusto.

Per poter stabilire questo occorre fornire una qualche misura delle prestazioni. In particolare è importante definire il come e il quando eseguire la valutazione delle azioni. Perciò si rende necessario un modello di successo, stabilito a priori, con il quale confrontare i risultati ottenuti. [8]

È possibile raggruppare gli agenti in 5 classi in base al grado di intelligenza percepita e alle capacità: [8]

- agenti con riflessi semplici (detti anche puramente reattivi);
- agenti con riflessi basati su modello;
- agenti basati su obiettivo;
- agenti basati su utilità;
- agenti che apprendono;

Per questo lavoro andremo a considerare la prima categoria riportata nell'elenco, ovvero agenti con riflessi semplici. Questa funzione d'agente infatti si comporta bene nel caso in cui l'ambiente sia completamente osservabile, come appunto sarà possibile fare in questo caso.

2.2 Sistema

Prima di comprendere il termine "sistema complesso" bisogna andare a capire cos'è un sistema. Esistono varie definizioni di questo termine, che dipendono principalmente dal contesto in cui viene utilizzato. Generalmente però si può definire un sistema come un gruppo di oggetti interagenti oppure indipendenti che fanno parte di un tutt'uno. Notare che i limiti di un sistema non sono sempre ben definiti.

Si può catalogare un sistema in base anche alla sua complessità strutturale ed operativa, come ad esempio nelle seguenti tre categorie:

- Sistema semplice: conosciuto e basato su causa-effetto (bicicletta)
- Sistema complicato: richiede conoscenza ma è studiabile (macchina)
- Sistema complesso: non conosciuta e che richiede ricerca, simulazione e analisi (comportamento di un organismo)

2.3 Sistema complesso

Innumerevoli definizioni possono essere associate al termine "sistema complesso"; in generale si può dire che un sistema complesso è un sistema strutturato, difficile sia da comprendere sia da testare, in cui ci sono diverse interazioni tra le componenti al suo interno. Inoltre, un sistema complesso è molto sensibile ad eventuali variazioni della situazione iniziale. Uno degli scopi principali per cui si realizzano questi sistemi è la simulazione, e alcuni esempi di sistemi complessi sono: [6]

- Automi cellulari
- Crosta terrestre e interazioni che provocano i terremoti
- Sistema climatico
- Sistemi sociali
- Sistemi economici
- Ambiente e territorio

2.4 Simulazione sistema

Rappresenta un modo di sfruttare modelli computazionali con lo scopo di valutare la scelta di piani, design, teorie e/o modelli. Spesso è necessario ricorrere ad un ambiente sintetico perchè il sistema reale può non essere facilmente osservabile.

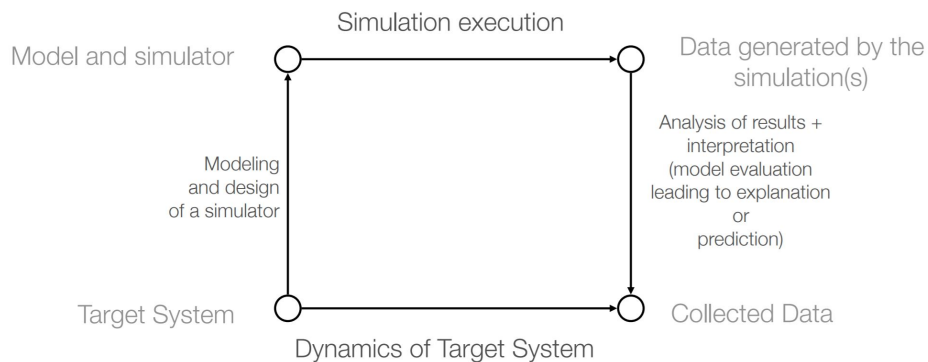


Figure 2: Ciclo di vita di una simulazione

2.5 Sistema multiagente

Un sistema multiagente o (sistema ad agenti multipli) è un insieme di agenti situati in un certo ambiente ed interagenti tra loro mediante una opportuna organizzazione. Un agente è cioè un'entità caratterizzata dal fatto di essere, almeno parzialmente, autonoma, sia essa un programma informatico, un robot, un essere umano, e così via.

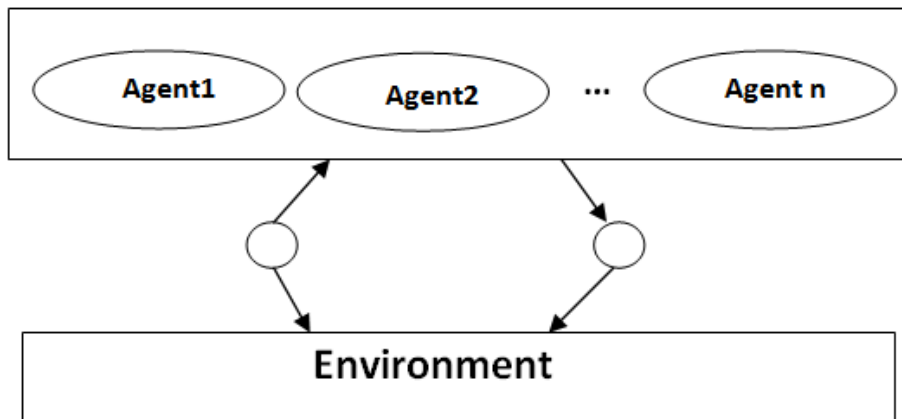


Figure 3: Interazione di più agenti con l'ambiente

Oggetto di ricerche da lunga data in intelligenza artificiale, i sistemi ad agenti multipli costituiscono un'interessante tipologia di modellazione di società, e hanno a questo riguardo vasti campi d'applicazione, che si estendono fino alle scienze umane e sociali (economia, sociologia, etc.). [7]

Un sistema multiagente, per essere definito tale, deve rispettare le seguenti caratteristiche: [12]

- Autonomia: ogni agente deve essere almeno parzialmente indipendente.
- Vista locale: nessun agente deve avere la vista globale del sistema.
- Decentralizzazione: nessun agente può controllare l'intero sistema.

2.5.1 Modelli di interazione tra agenti

Due o più agenti interagiscono tra loro se sono in una relazione dinamica attraverso sequenza di azioni reciproche. Due agenti sono portati ad interagire tra di loro quando non è possibile effettuare o portare a termine un determinato compito da parte di un agente singolo. Ciò può accadere a causa di risorse insufficienti, abilità insufficienti oppure per un approccio distribuito.

Esistono diversi modelli di interazione tra agenti, che possono essere suddivisi a loro volta in due grandi categorie: modelli di interazione diretta e modelli di interazione indiretta.

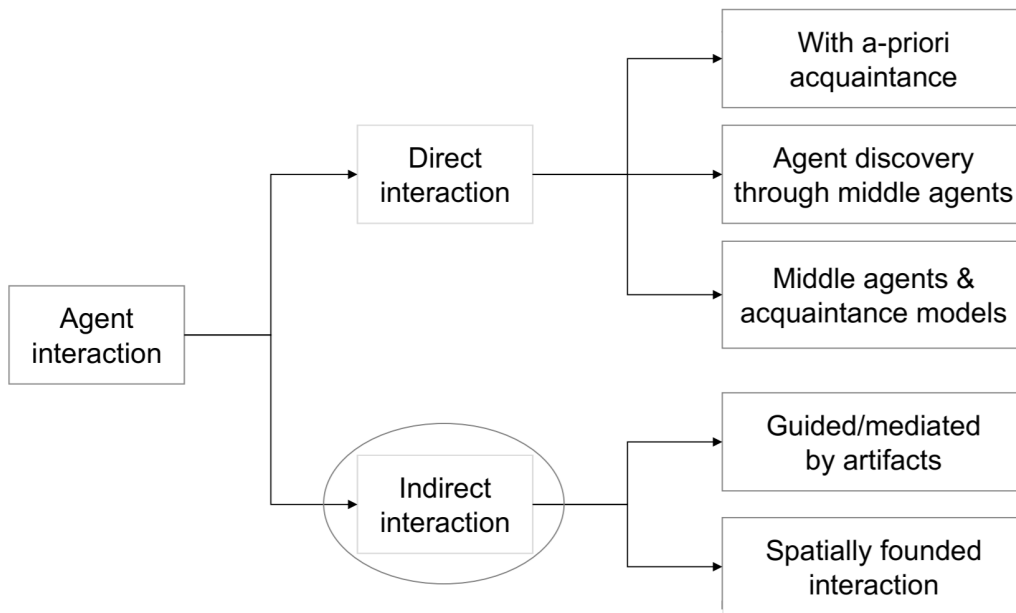


Figure 4: Modelli di interazione tra agenti

Nei modelli ad interazione diretta gli agenti sono in grado di comunicare direttamente tra di loro. Lo scambio di informazioni è reso possibile grazie all'utilizzo di un agent communication language (ACL) e di una struttura dei messaggi. La comunicazione tra due agenti di questo genere avviene quindi in maniera indiscriminata dato che l'informazione va dal mittente al destinatario senza alcun tipo di dispersione e/o intermediario esterno.

Questi modelli sono facilmente implementabili e simili a sistemi distribuiti già

esistenti, anche se allo stesso tempo sono presenti regole strette sul metodo di comunicazione e ogni agente deve essere a conoscenza di un altro agente prima di poterci comunicare.

Nei modelli ad interazione indiretta invece gli agenti interagiscono attraverso un ente che funziona di intermediario. Questo ente fornisce dei meccanismi di comunicazioni e delle regole di accesso. L'implementazione di un modello del genere permette di avere un'interazione mediata, e quindi maggior possibilità di controllo, seppur l'implementazione risulta molto complessa per alcuni contesti e alcuni sistemi distribuiti.

2.5.2 Tipologie di ambienti

Un sistema multiagente consiste negli agenti e del relativo ambiente. Le diverse categorie di agenti sono già state riportate nella sezione precedente 2.1, mentre per quanto riguarda l'ambiente può essere suddiviso in:

Accessible E' possibile conoscere tutte le informazioni riguardanti le possibili combinazioni dell'evoluzione dell'ambiente.	Inaccessible Non è possibile conoscere tutte le informazioni riguardanti le possibili combinazioni dell'evoluzione dell'ambiente.
Deterministic E' possibile conoscere in maniera deterministica tutti i cambiamenti che una certa azione porta al sistema.	Non Deterministic Non è possibile conoscere in maniera deterministica tutti i cambiamenti che una certa azione porta al sistema.
Episodic Le performance di un agente dipendono dal numero di episodi discreti avvenuti, con nessun rapporto con le performance dell'agente in uno scenario diverso.	Non Episodic Le performance di un agente non dipendono dal numero di episodi discreti avvenuti, e l'agente prende decisioni su scenari differenti.
Static Non è soggetto a variazioni sul suo stato durante il corso della finestra temporale in cui l'agente sta agendo.	Dynamic E' soggetto a variazioni sul suo stato durante il corso della finestra temporale in cui l'agente sta agendo.
Discrete E' possibile rappresentare con una cardinalità finita gli stati dell'ambiente.	Continuous Non è possibile rappresentare con una cardinalità finita gli stati dell'ambiente.

Table 1: Diverse tipologie di ambiente

2.6 Automated Guided Vehicle

Un veicolo a guida autonoma (AGV) identifica dei veicoli, solitamente utilizzati in campo industriale per il trasporto di prodotti da un punto dello stabilimento ad un altro.



Figure 5: Diversi modelli di AGV

Esistono numerose tecnologie adottate per la realizzazione di AGV, ognuna con vantaggi e svantaggi; la scelta tecnica di realizzazione infatti è fortemente correlata al problema e all'ambiente che si sta considerando. degli AGV:

Navigation	Path decision
<ul style="list-style-type: none"> - Wired - Guide tape - Laser target navigation - Inertial navigation - Natural features navigation - Vision guidance - Geoguidance 	<ul style="list-style-type: none"> - Frequency select mode - Path select mode - Magnetic tape mode
Traffic control	Battery charging
<ul style="list-style-type: none"> - Zone control - Collision avoidance - Combination control 	<ul style="list-style-type: none"> - Battery swap - Automatic and opportunity charging - Automatic battery swap

Table 2: Diverse tipologie di tecnologie utilizzate per realizzazione AGV

2.7 Automated Guided Vehicle System

Un sistema automatizzato di veicoli a guida autonoma (AGVS) permette una gestione estremamente flessibile per spedizione e gestione di materiali, soprattutto per quegli ambienti in cui i prodotti sono molto vari e le necessità continuano a variare nel tempo.

Questi sistemi sono stati introdotti nel 1950, e da allora si sono trovati sempre più contesti e applicazioni in cui forniscono un'elevata utilità pratica. Inoltre, con l'avvento di tecnologie sempre più avanzate e di MAS sempre più sofisticati, le performance di questi sistemi sono in continua crescita.

2.7.1 Classificazione di un AGVS

Un AGVS deve essere progettato e implementato considerando il contesto in cui ci si trova e le priorità che si presentano. Tutto sommato, un AGVS può essere classificato in base ai tre seguenti requisiti:

- **Percorso di guida:** Può essere statico, in cui la navigazione avviene seguendo una serie di percorsi prestabiliti (unidirezionale o bidirezionale) oppure può essere dinamico, in cui la navigazione dei veicoli avviene in maniera completamente autonoma.
- **Capacità del veicolo:** Un AGV può avere una capacità di carico diversa da un altro, come ad esempio carico unitario e carico multiplo.
- **Indirizzamento dei veicoli:** L'indirizzamento dei veicoli può essere indiretto, in cui un AGV può visitare solamente determinate stazioni, oppure diretto, in cui ogni AGV può visitare ogni stazione.

2.7.2 Performance di un AGVS

Solitamente la valutazione delle performance di un AGVS avviene tramite simulazione che, attraverso la raccolta di parametri rilevanti, permette di misurare le performance di un sistema rispetto ad un altro. Esempi di parametri di produzione che possono essere valutati di un AGVS sono:

- Ordini svolti
- Tempo medio di attesa per ordini pronti

- Coda di ordini in attesa
- Numero di conflitti
- Numero di deadlock
- Numero totale di AGV impiegati
- Tempo totale di utilizzo degli AGV
- Tempo in cui un AGV è impiegato di un azione
- Tempo in cui un AGV non è impiegato in nessuna azione

2.8 Lavori precedenti

Nonostante svariati lavori sono stati già ampiamenti portati a termine e documentati non è risultato del tutto facile ed intuitivo trovare dei lavori su cui basare le nostre scelte implementative e progettuali.

Nonostante ciò alcuni lavori ci sono tornati estremamente utili per delle sottosezioni del progetto; ad esempio il lavoro documentato all'interno dell'articolo scientifico "Evaluation of automatic guided vehicle systems" [5] ci ha permesso di individuare e definire le metriche utili alla valutazione del nostro simulatore. Un altro articolo scientifico di estrema utilità è risultato essere "Multi Agent Simulation for Decision Making in Warehouse Management" [4], in cui viene analizzato e simulato un sistema su un ambiente molto simile a quello affrontato da questo studio.

3 Descrizione del dominio

3.1 Azienda

LDE s.r.l nasce nel 1997 come azienda per il trasporto e la logistica specializzata in abiti e tessuti. Ad oggi conta circa 30 dipendenti e un fatturato di oltre un milione di euro. Si è entrati in contatto con questa azienda grazie ad una collaborazione passata.

3.2 Tipologia di lavoro e di ordini

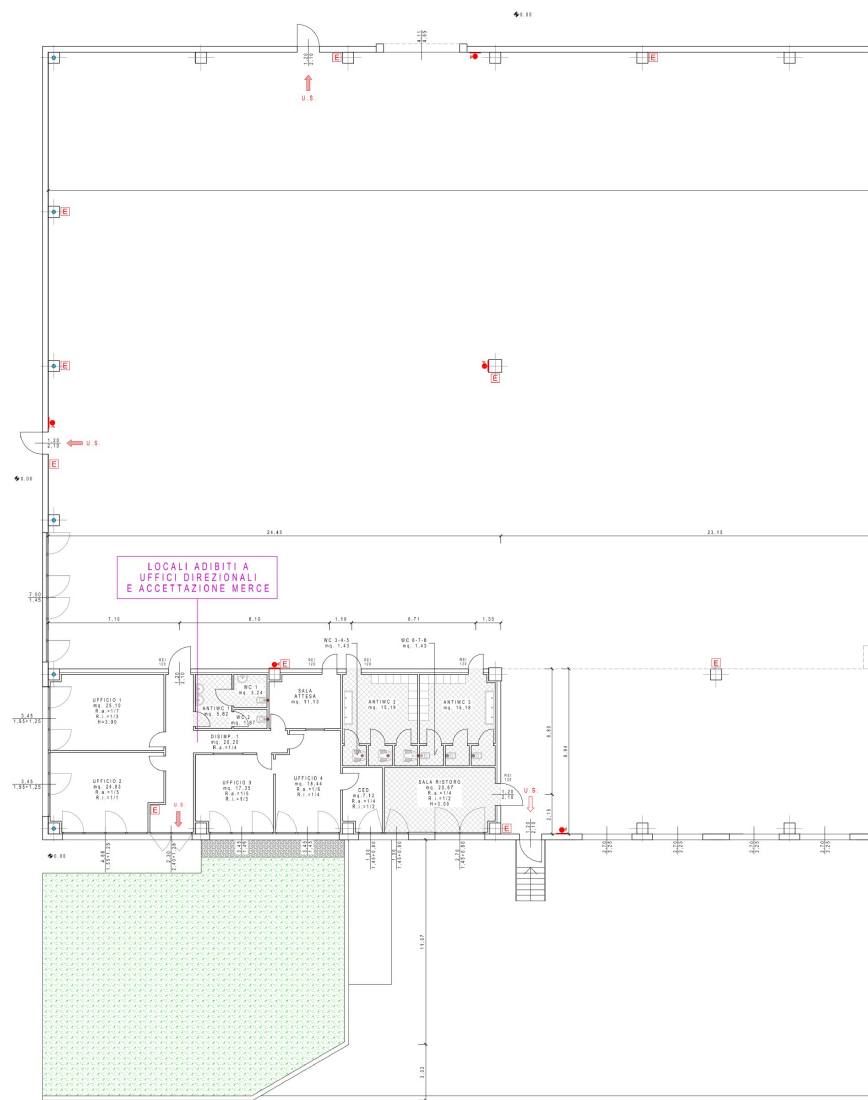
Il lavoro svolto da questa azienda consiste principalmente nell'elaborazione, preparazione, ricezione e spedizione di articoli. Gli articoli in questione possono essere di svariati tipi, dai vestiti agli accessori, da macchinari a utensili da cucina,.. In questo studio si andrà a considerare la parte del magazzino che tratta gli ordini inerenti agli articoli in ambito vestiario da loro trattati.

3.3 Ambiente

L'ambiente del magazzino preso da noi in considerazione consta di una zona per gli uffici (in basso a sinistra), 4 gate per caricare i camion (di cui uno chiuso, motivo per cui ne sono stati rappresentati tre sulla mappa della simulazione) e 8 corsie in cui sono posizionate le merci. Nella simulazione, tutto è rappresentato in scala 1:1 con piccole approssimazioni che non influiscono né sulle performance, né sull'aderenza alla realtà della simulazione. Per quanto riguarda l'ambiente, le uniche assunzioni che sono state fatte riguardano gli oggetti e, in particolare:

- Gli oggetti nel magazzino sono sempre disponibili per un AGV che li richiede.
- Un AGV prende un oggetto posizionandosi in un punto preciso per la ricezione di quell'oggetto che viene posizionato sull'AGV da un ipotetico braccio meccanico.

Di seguito è riportata la cartina rappresentante la perimetria del dominio in questione, che verrà utilizzata nella fasi successive per ottenere un ambiente di simulazione verosimile a quello reale.



3.4 Lista degli ordini

Insieme alla cartina del magazzino è stato possibile reperire un elenco di ordini che vengono svolti quotidianamente; la lista di cui si sta parlando contiene 2023 ordini differenti. Ciascun ordine contiene una lista di articoli che può variare e un unico destinatario, che può essere:

- Milano (MI)
- Firenze (FI)
- Como (CO)

Il seguente grafico riporta il numero di ordini presenti all'interno della lista per ciascuna destinazione.

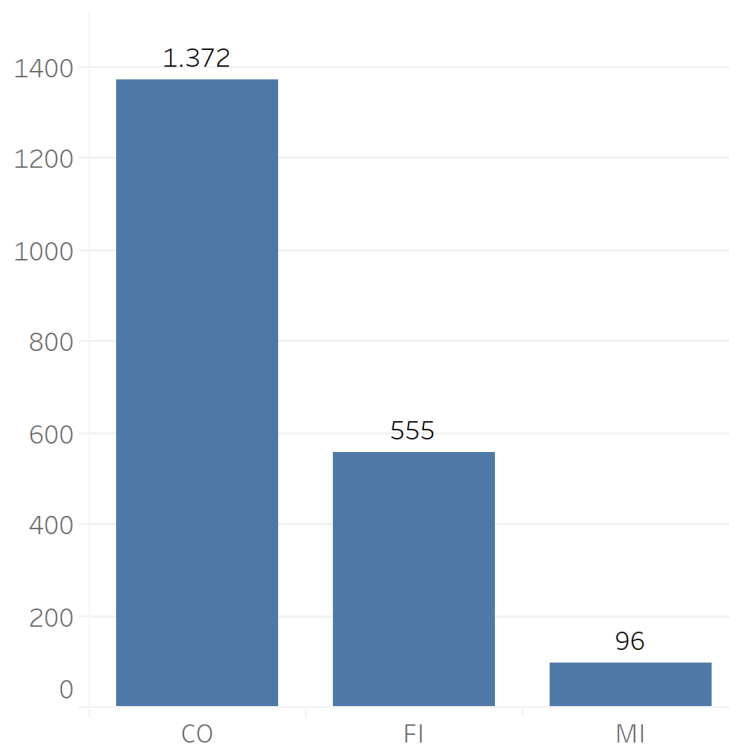


Figure 7: Numero di ordini per ciascun cliente

4 Descrizione del modello

4.1 Ambiente del simulatore

Una volta ottenuta la mappa del magazzino, si è andati a cercare di rappresentare la struttura del magazzino in maniera tale da renderla utilizzabile per un sistema di simulazione.

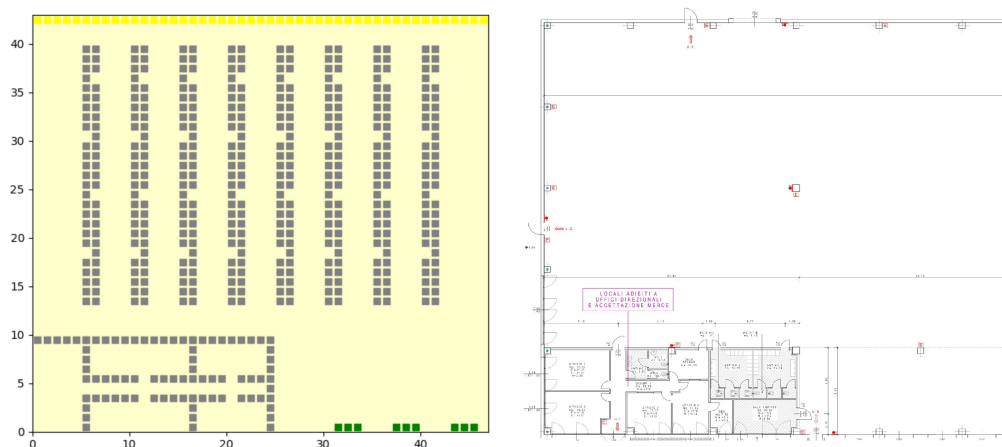
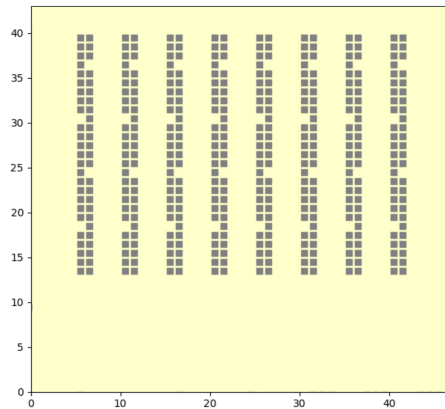


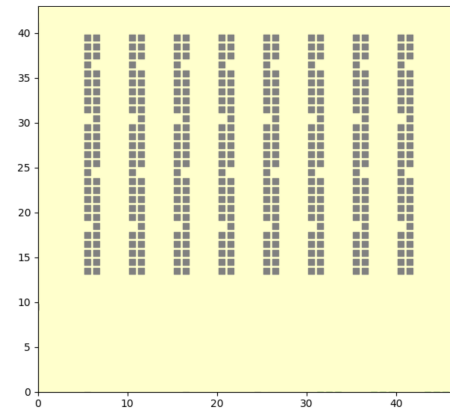
Figure 8: Rappresentazione della mappa del magazzino sul simulatore

All'interno dell'ambiente sono stati definiti i seguenti elementi, di cui i primi 4 sono quelli fondamentali alla simulazione del problema e meglio descritti nella pagina seguente:

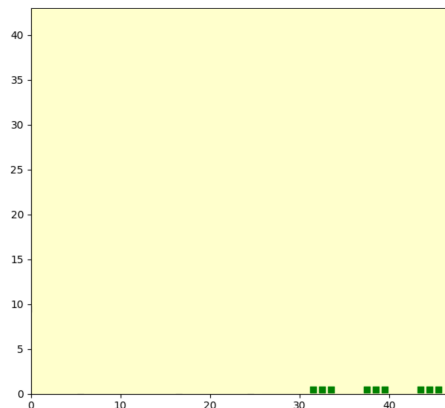
- Scaffalature
- Punti di carico
- Punti di scarico
- Area di attesa
- Area di ricarica
- Uffici:



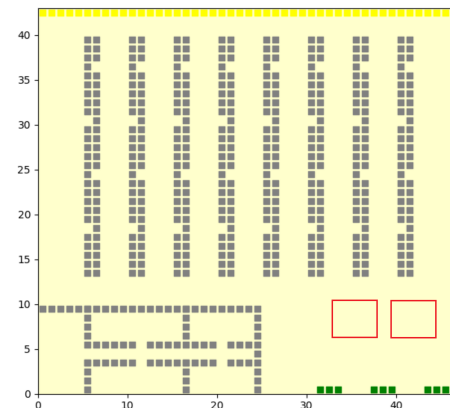
Scaffalature: Rappresentate con il colore grigio al centro dell'ambiente di simulazione, rispecchiando le corrispettive 8 scaffalature presenti all'interno del magazzino fisico.



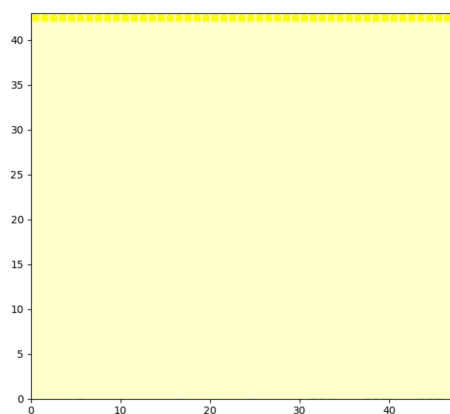
Punti di carico: Rappresentati con delle interruzioni della corsia sono collocati lungo le corsie stesse, dove l'AGV potrà accedere per poter prelevare le merci richieste.



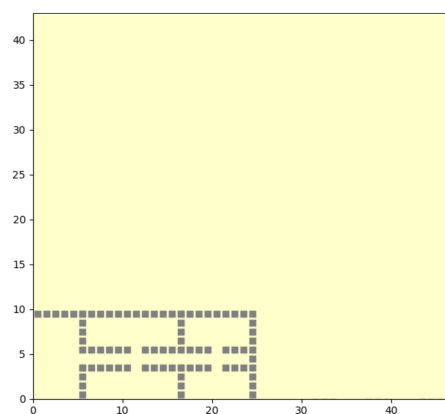
Punti di scarico: Rappresentati con il colore verde. Questi sono i punti in cui verranno rilasciate le merci prelevate dagli AGV in precedenza.



Area di attesa: Indicata con due rettangoli rossi di fronte ai punti di scarico. Gli AGV in attesa di scaricare attenderanno qui il loro turno.



Area di ricarica: Rappresentate con il colore giallo in alto all'interno dell'ambiente di simulazione. Durante questo lavoro sono state utilizzate solamente come punto di partenza degli AGV per la simulazione e per possibili lavori futuri.



Uffici: Rappresentati con il colore grigio in basso a sinistra all'interno dell'ambiente di simulazione. Non hanno uno scopo effettivo bensì sono stati riportati per rimanere coerenti con la mappa originale del magazzino e non considerare spazi effettivamente non disponibili.

4.2 Modellazione agenti

Ciascun agente definito all'interno della simulazione è caratterizzato da i seguenti attributi, che permettono la corretta gestione del sistema di simulazione e del comportamento degli agenti stessi:

- **Id:** un valore intero $[1, 12]$ univoco associato a ciascun agente presente nella simulazione, permettendo il riconoscimento di ciascuno di essi.
- **Colore:** un valore alfanumerico ["black"] che permette di rappresentare gli agenti con colori differenti all'interno della simulazione. Il simulatore attuale riporta tutti gli agenti con il colore nero.
- **Posizione:** una coppia di numeri interi (x, y) che rappresenta la posizione dell'agente all'interno dell'ambiente di simulazione. Ciò permette di tener traccia della posizione di tutti gli agenti in qualsiasi momento della simulazione.
- **Posizione iniziale:** una coppia di numeri interi (x, y) che rappresenta la posizione da cui l'agente è partito all'inizio della simulazione. Verosimilmente questa posizione dovrebbe coincidere con la posizione di una stazione di ricarica, e permetterebbe all'agente, una volta raggiunto il suo obiettivo, di tornare alla sua postazione di ricarica prestabilita.
- **Stato:** un valore alfanumerico che permette di capire cosa sta effettivamente facendo un certo agente. Questo attributo è fondamentale per la corretta elaborazione degli ordini e per lo svolgimento della simulazione. Nei seguenti paragrafi sono riportati gli stati assumibili da un agente [4.2.1] e le corrispettive transazioni possibili tra di essi [4.2.2].
- **Info sull'ordine:** un valore intero $[-1, 200]$ che indica il numero dell'ordine su cui sta lavorando l'agente in quel determinato momento. Se il valore è pari a -1 vuol dire che l'agente non sta lavorando su nessun ordine in quel momento.
- **Cliente:** un valore alfanumerico ["MI", "FI", "CO"] che indica la sigla del cliente dell'ordine che sta effettuando in un determinato momento. Ciò permette di determinare dove l'agente dovrà andare a scaricare l'ordine.

- **Percorso:** una lista di coppie di valori interi $[(x1, y1), (x2, y2), \dots, (xN, yN)]$ che rappresenta il percorso calcolato e previsto dall'agente per raggiungere una determinata posizione (xN, yN) all'interno dell'ambiente di simulazione .
- **Gate:** un valore alfanumerico $[-1, 0, 1, 2]$ che rappresenta il gate a cui è diretto l'agente in quel momento. Ciascun gate è associato ad un determinato cliente, mentre il valore -1 significa che in quel momento l'agente non si sta dirigendo verso nessun gate. Ciò permette al sistema di gestione di controllare il flusso d'arrivo ai gate dei vari agenti.
- **Goal:** un coppia di valori interi (x, y) che rappresenta la posizione all'interno dell'ambiente di simulazione che l'agente deve raggiungere in quel momento per compiere un'azione qualsiasi.
- **Articoli prioritari:** una lista di valori alfanumerici $["Camicia", "Scarpa arancio", \dots, "Giubbetto"]$ che rappresenta il nome degli articoli che rappresenteranno una priorità assoluta per gli agenti in questione. Ciò permette di indirizzare un agente verso l'elaborazione di ordini contenuti questi articoli rispetto ad altri ordini.

4.2.1 Stati assumibili

Come accennato nel paragrafo precedente, ciascun agente è caratterizzato da svariati attributi, tra cui uno che definisce lo stato in cui si trova l'agente in un determinato momento. Questo attributo ha un'estrema importanza per il corretto funzionamento della simulazione, in quanto regola il comportamento degli agenti in base allo stato in cui si trovano.

Di seguito è riportata la lista dei possibili stati che un agente può assumere durante il ciclo di vita di una simulazione insieme ad una breve descrizione per ciascuno di essi:

- **Free:** un agente in questo stato non sta elaborando nessun ordine, rimanendo quindi disponibile ad essere impiegato per l'elaborazione di un ordine. Tutti gli agenti all'inizio di una simulazione si trovano in questo stato.
- **ToGoal:** un agente in questo stato ha in carico un determinato articolo da prelevare, quindi si starà muovendo verso il punto di prelievo associato all'articolo in questione.

- **Loading:** un agente in questo stato si trova fisicamente all'interno in uno dei punti di prelievi all'interno dell'ambiente di simulazione per caricare un determinato articolo. Una volta caricato l'articolo, prima di lasciare il punto di prelievo, calcola il percorso per arrivare al punto di scarico designato per l'articolo in questione.
- **ToGate:** un agente in questo stato ha in carico un articolo da scaricare presso un determinato gate, quindi si starà muovendo verso il punto di scarico del cliente designato.
- **ToWaitP:** un agente in questo stato ha in carico un articolo da scaricare presso un determinato gate ma, in quel determinato momento, il gate è occupato da altri agenti che stanno scaricando articoli, quindi si starà muovendo verso l'area di attesa.
- **Wait:** un agente in questo stato ha in carico un articolo da scaricare presso un determinato gate e si trova già all'interno di un'area di attesa dato che il gate interessato è attualmente occupato da altri agenti. Rimarrà quindi in attesa che si liberi un gate per poter andare a scaricare l'articolo prelevato.
- **Unloading:** un agente in questo stato si trova fisicamente all'interno in uno dei gate all'interno dell'ambiente di simulazione per scaricare un determinato articolo. Una volta scaricato l'articolo, prima di lasciare il gate, verifica la presenza di altri articoli da elaborare e, se esistono, calcola il percorso per arrivare al prossimo punto di prelievo altrimenti imposta il percorso per tornare al suo punto di parte iniziale.
- **ToHome:** un agente in questo stato ha completato gli ordini di sua competenza e si dirige verso la sua postazione di ricarica all'interno dell'ambiente di simulazione.
- **Home:** un agente in questo stato si trova fisicamente presso la sua postazione di ricarica all'interno dell'ambiente di simulazione. Tutti gli agenti alla fine di una simulazione si trovano in questo stato.

4.2.2 Transizioni tra stati

Nel paragrafo precedente [4.2.1] sono stati riportati e descritti tutti gli stati assumibili da un qualsiasi agente durante l'esecuzione di una simulazione. Il seguente schema rappresenta tutte le possibili transazioni tra gli stati di un agente, andando a dare una chiara idea del possibile ciclo di vita di un agente durante un'intera simulazione.

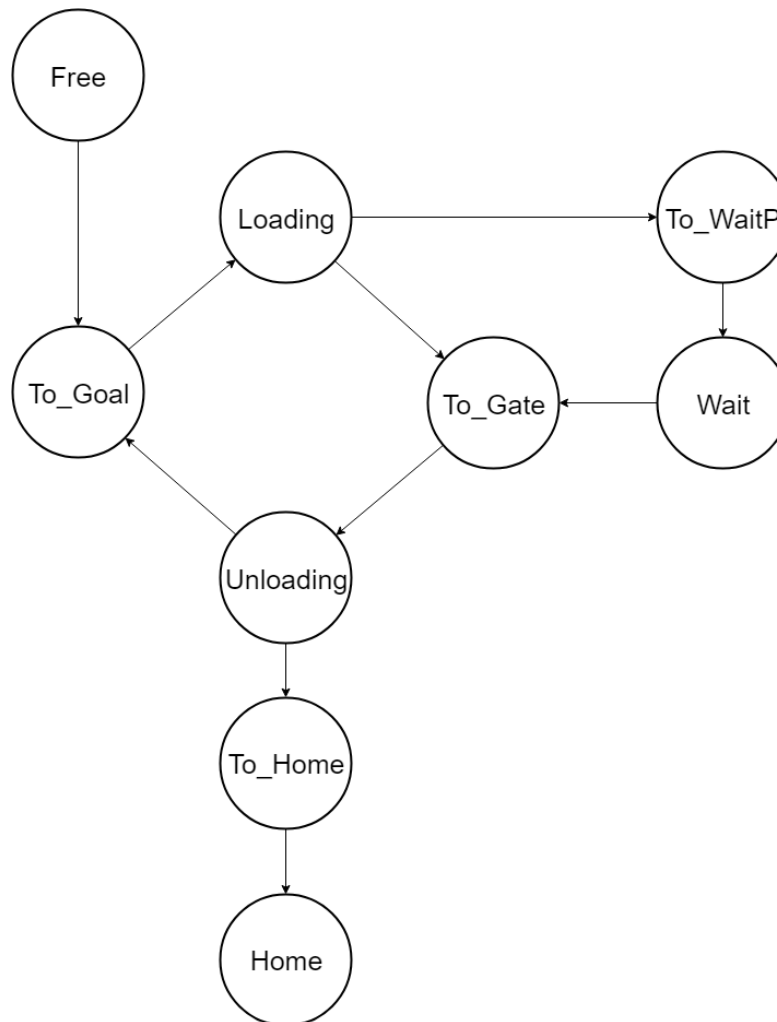


Figure 9: Transazioni tra stati di un AGV

4.3 Gestione navigazione

In questo lavoro si è deciso di sperimentare una navigazione libera da parte degli AGV, basandosi sulla loro capacità di visione attraverso sensori.

Si è scelto di prendere una decisione di questo genere per andare ad ottimizzare le distanze percorse dai singoli agenti, migliorando quindi durata della batteria, tempo di esecuzione dei task e usura delle componenti fisiche. Per fare ciò si è dovuto assumere che un sistema del genere non possa prevedere la presenza simultanea di impiegati umani e AGV all'interno del magazzino.

L'algoritmo utilizzato dagli AGV per la navigazione all'interno dell'ambiente di simulazione è l'algoritmo di Lee [9]. Questo algoritmo, basato su BFS [10], viene solitamente utilizzato per calcolare il percorso da un punto ad un altro di un labirinto.

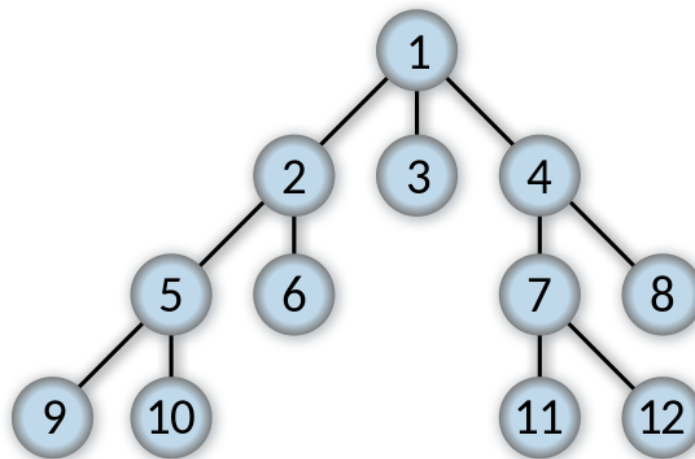
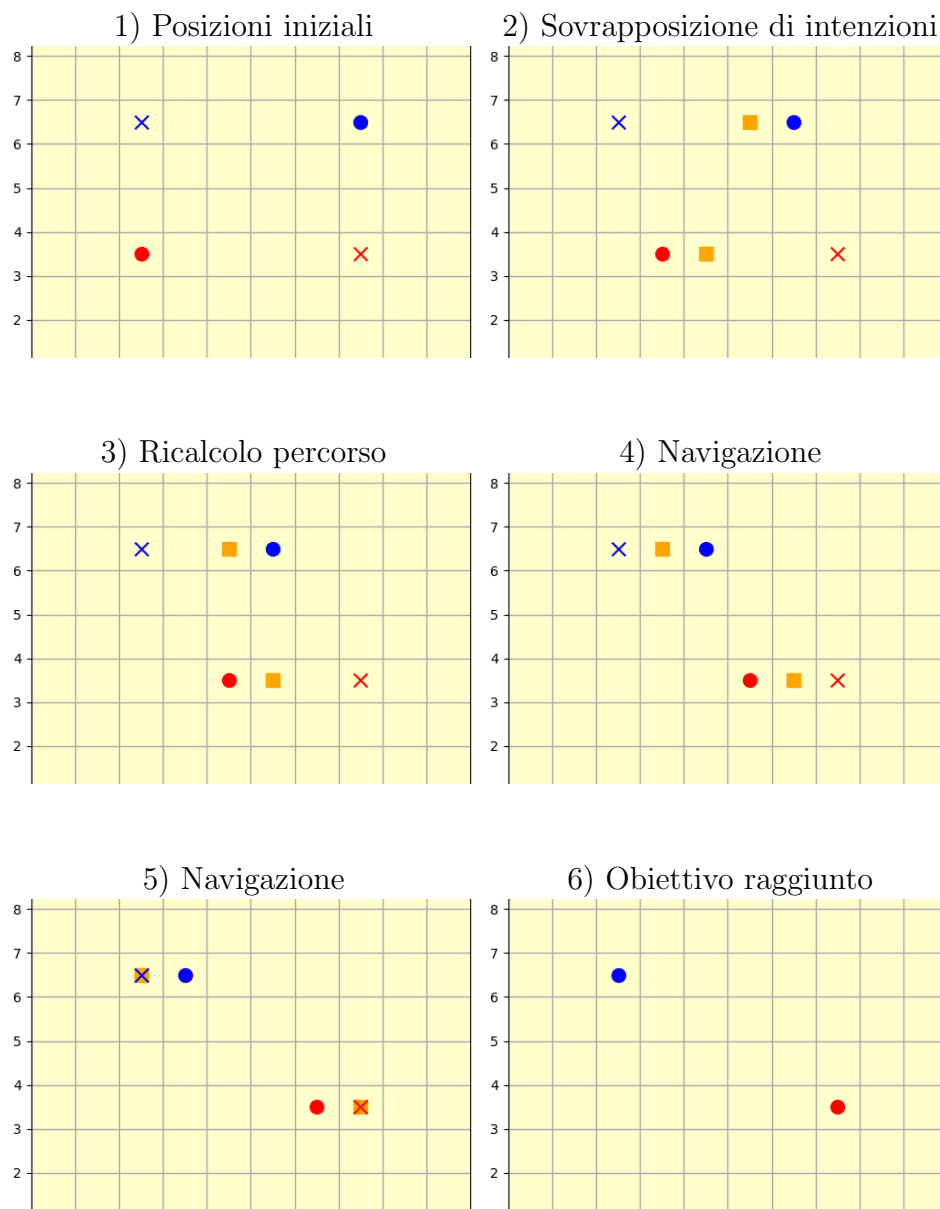


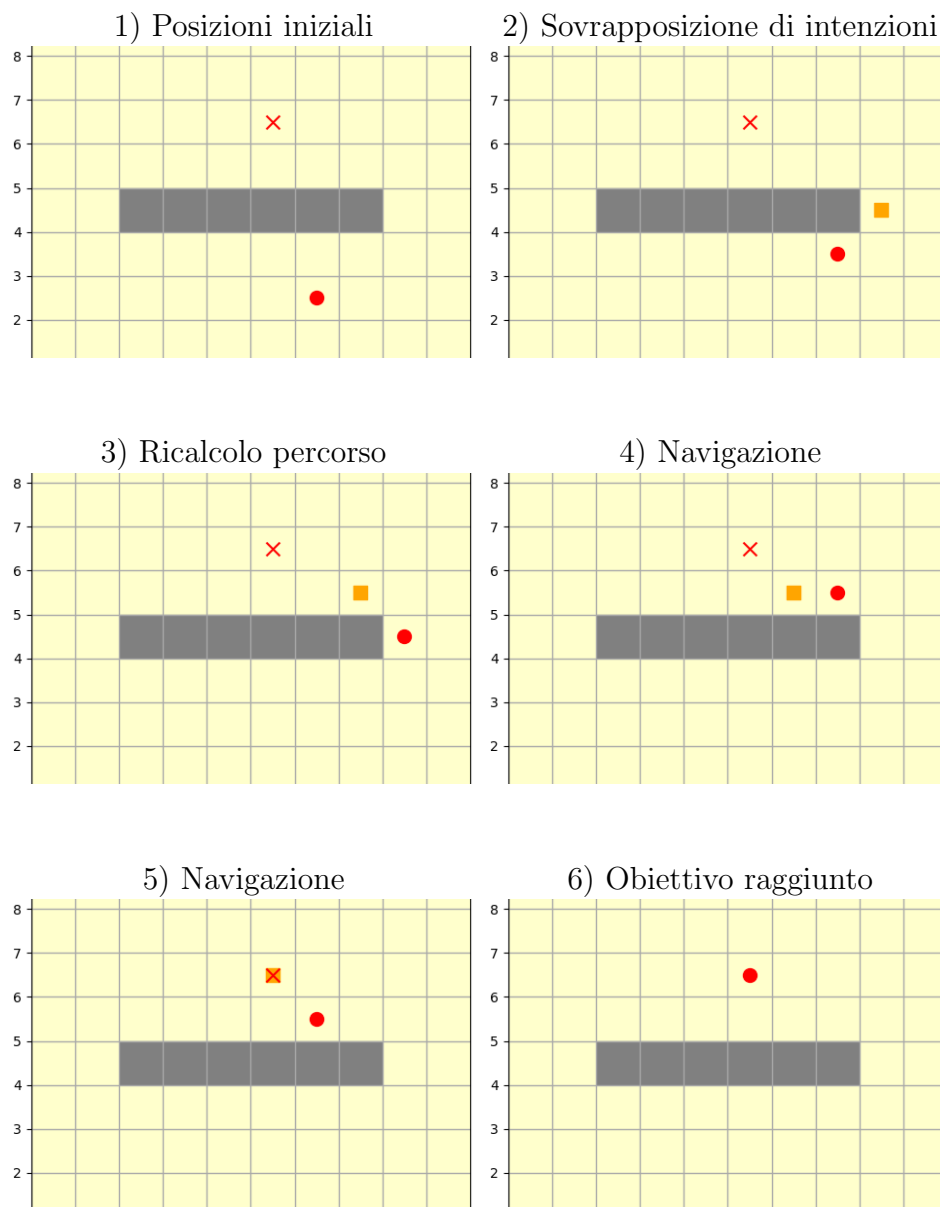
Figure 10: BFS: Ordine in cui i nodi vengono esplorati

Permette infatti di ottenere sempre la soluzione migliore (se esiste), ma allo stesso tempo può essere lento e dispendioso in termini di memoria. Si è deciso di utilizzare proprio questo algoritmo rispetto ad altri perchè l'ambiente di simulazione trattato in questo caso presenta delle dimensioni così ridotte che non è necessario andare ad ottimizzare la computazione in termini di tempo e memoria, andando ad ottenere sempre la soluzione ottima senza richiedere una mole computazionale significativa.

4.3.1 Esempio navigazione - Libero



4.3.2 Esempio navigazione - Ostacolo



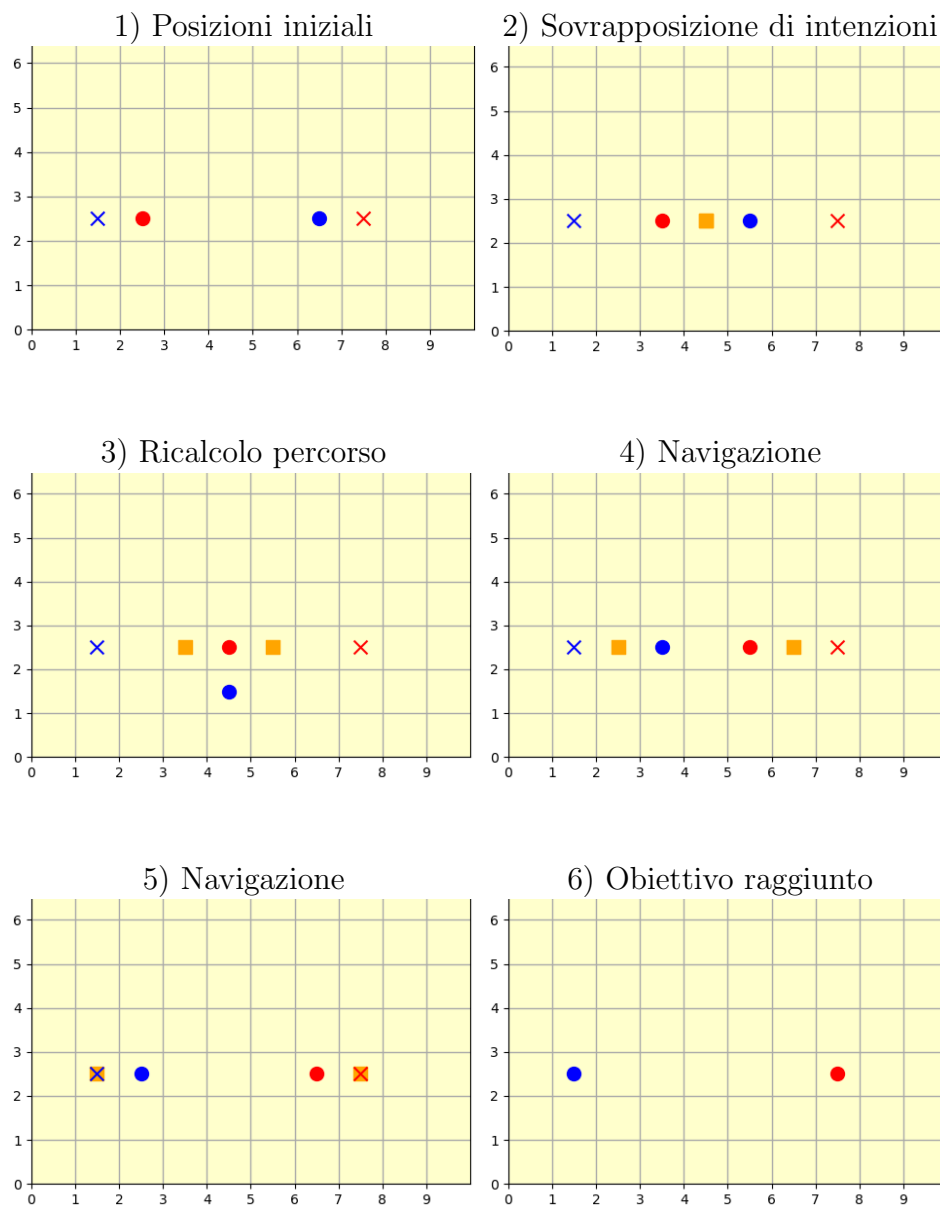
4.4 Gestione conflitti

Il sistema di navigazione sopra descritto permette agli agenti di navigare in maniera ottimale su una mappa completamente statica, ciò che non varia nessun tipo di variazioni nel corso del loro percorso. Ovviamente un algoritmo di navigazione del genere porterebbe diverse problematiche, dato che durante la simulazione ci sono altri agenti che si muovono per l'ambiente. Proprio per questo motivo è stato necessario andare a progettare e realizzare un gestore dei conflitti di navigazione, in modo da risolvere qualsiasi problema potessero incontrare gli agenti durante il loro percorso.

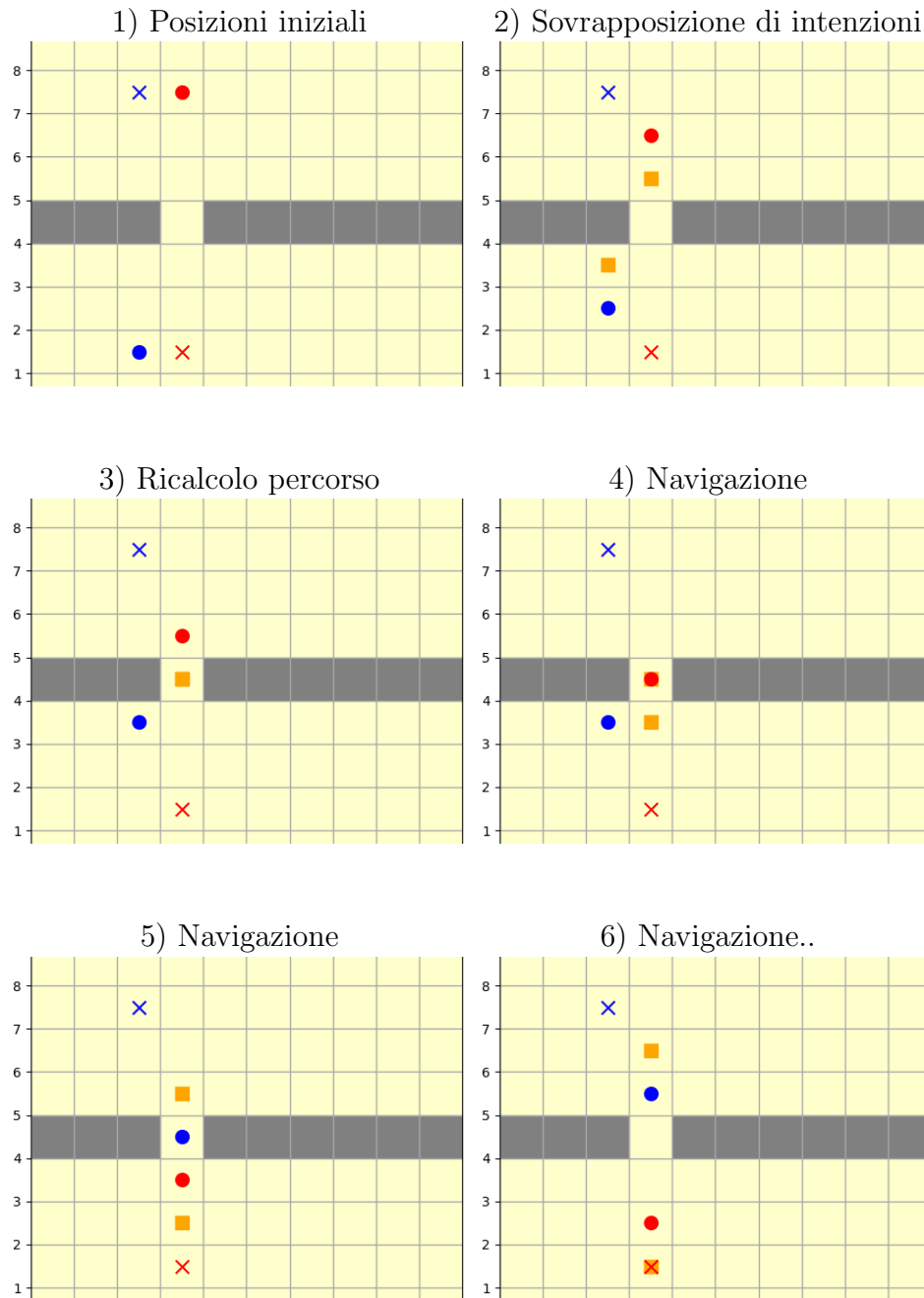
Si definisce come "conflitto" la situazione in cui un agente si trova ad avere un ostacolo imminente sul suo percorso designato in precedenza, obbligandolo a dover ricalcolare il percorso. Le possibili soluzioni a questo problema sono principalmente due:

- L'agente sta fermo per uno step temporale, andando a controllare se lo step successivo persiste ancora lo stesso conflitto o meno.
- L'agente ricalcola il percorso per arrivare alla sua destinazione, aggiungendo la posizione dove si è presentato il conflitto alla lista di posizioni non considerabili per la navigazione.

4.4.1 Esempio conflitto - Cambio direzione



4.4.2 Esempio conflitto - Attesa



4.5 Gestione area scarico merci

Come rappresentato nel simulatore [4.1] sono presenti tre punti di scarico detti "gates". Queste locazioni sono dove gli AGV devono scaricare gli articoli prelevati all'interno del magazzino. Considerando che gli ordini in input sono indirizzati verso tre destinazioni (MI, FI, CO), si è ipotizzato e deciso di assegnare a ciascun gate una specifica destinazione.

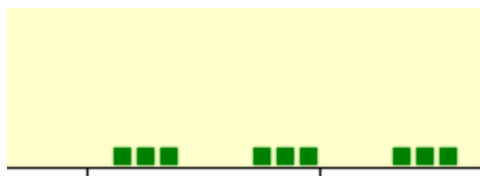


Figure 11: Area scarico merci: MI, FI, CO

Inoltre, ciascun gate presenta solamente tre aree di scarico merci per gli AGV: una volta che una di queste aree di scarico merci viene occupata da un certo ordine, quest'ultimo deve essere portato a termine prima di poter essere nuovamente occupata con un nuovo ordine.

4.6 Gestione area di attesa

Si è deciso di collocare due zone di attesa di fronte ai gates, ciascuna delle quali ha sei postazioni di attesa per AGV. Quindi un agente entrerà in attesa se non sono presenti zone di scarico disponibili per il cliente associato all'articolo che sta elaborando.

L'uscita dalla fase di attesa è determinata esclusivamente dalla disponibilità di un punto di scarico associato al cliente interessato. Se dovesse capitare che più AGV in attesa concorrono per la stessa postazione di scarico che si è appena liberata verrà applicato il meccanismo FIFO (First In First Out), che permetterà agli agenti in attesa da maggior tempo di riprendere l'elaborazione dei propri articoli in maniera corretta.

4.7 Gestione ordini e agenti

4.7.1 Behaviour type 1

Questo metodo fa sì che ciascun AGV, una volta preso in carico un ordine, lo inizierà e lo porterà a termine in maniera autonoma. Non è previsto nessun tipo di collaborazione e/o concorrenza. La presa in carico degli ordini è puramente sequenziale.

4.7.2 Behaviour type 2

Questo metodo fa sì che ciascun AGV possa elaborare solamente un certo sottoinsieme degli articoli presenti all'interno del magazzino, come ad esempio una singola corsia oppure una categoria di articoli. Succederà quindi che un AGV andrà a lavorare su tutti quegli ordini che necessitano uno o più articoli di quelli da lui processabili. Ciò permette di ottenere un buon livello di collaborazione tra agenti, in quanto ciascuno si occupa di una parte dell'ordine, ma crea anche degli episodi di concorrenza per quanto riguarda la condivisione delle porte di scarico.

4.7.3 Behaviour type 3

Questo metodo fa sì che ciascun AGV possa elaborare un qualsiasi articolo di un qualsiasi ordine in maniera puramente sequenziale. Un AGV andrà quindi a prendere il primo ordine in cima alla lista degli ordini da svolgere e andrà ad elaborare il primo degli articoli che necessitano ancora. Questo metodo di gestione crea una forte collaborazione tra agenti che, in base alle dimensioni di un ordine e non agli articoli al suo interno, possono collaborare tutti insieme su un singolo ordine.

5 Implementazione simulatore

5.1 Interfaccia simulatore

Il sistema di simulazione è stato interamente sviluppato tramite il linguaggio di programmazione Python, andando ad usufruire in particolare del modulo PycxSimulator. Di seguito è riportata la struttura della schermata durante il ciclo di vita di una simulazione. I tasti sulla destra permettono all'utente di incominciare, finire o mettere in pausa la simulazione, mentre i grafici sulla sinistra permettono di monitorare la simulazione stessa e i relativi dati.

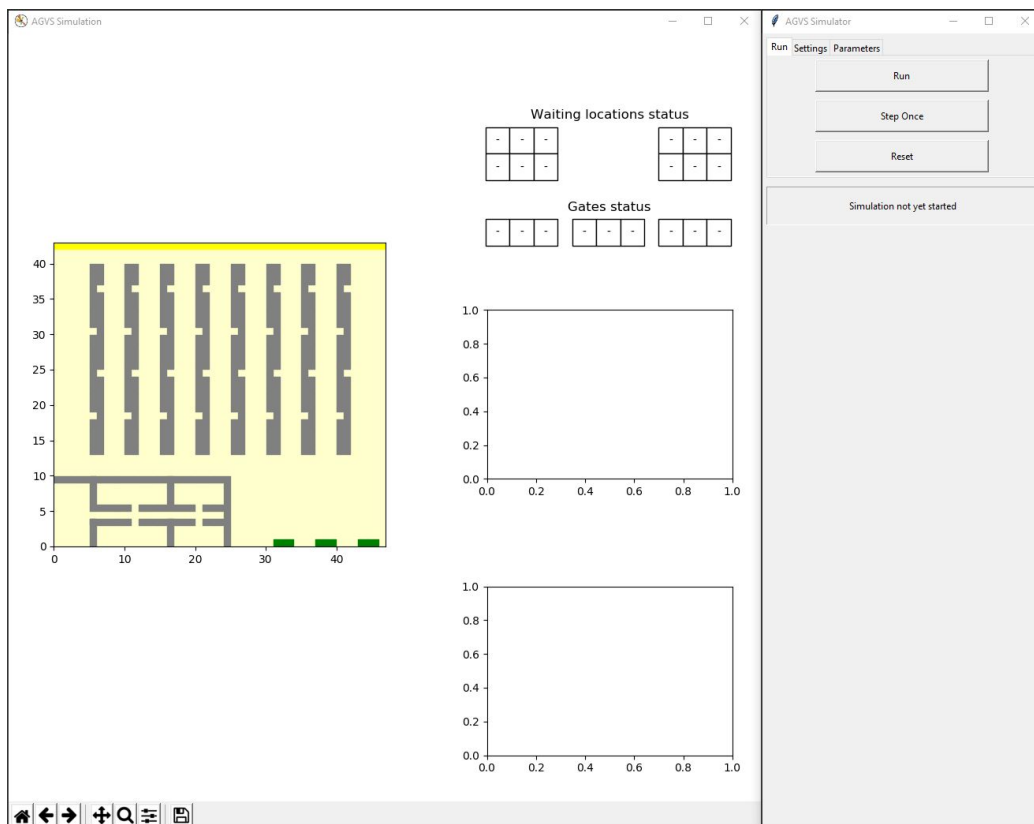
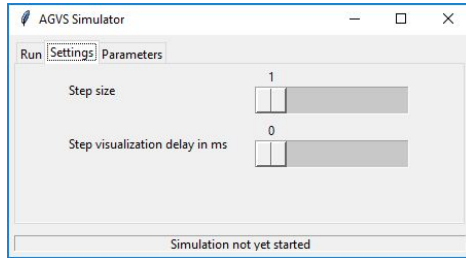
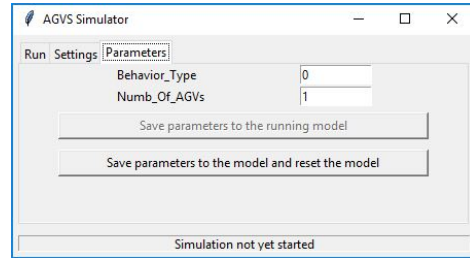


Figure 12: GUI del simulatore implementato

Settings: dalla tab "settings" della schermata di gestione è possibile scegliere con quale intervallo di step temporali far aggiornare i grafici e l'ambiente relativi alla simulazione.



Parameters: dalla tab "parameters" della schermata di gestione è possibile scegliere con quale parametri avviare la simulazione (Behaviour type e numero di AGV) tra quelli predefiniti:



Durante l'esecuzione di una simulazione, come già detto in precedenza, i grafici riportati nella GUI del simulatore permettono all'utente di monitorare il ciclo di vita e lo sviluppo di una simulazione. L'immagine 13 riporta un esempio di tutto ciò che viene riportato graficamente durante il processo.

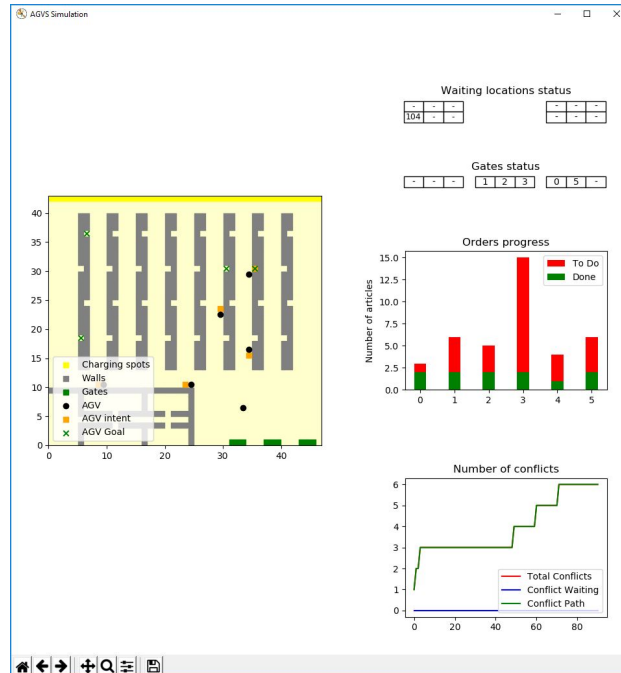


Figure 13: GUI del simulatore durante esecuzione

6 Simulazione

Dopo aver progettato l'AGVS ed implementato il sistema di simulazione, si è passati alla fase di simulazione. Questa fase si pone l'obiettivo di simulare il funzionamento degli agenti con diverse configurazioni di comportamento, tenendo invariato il set di ordini in input. Ciò fornisce la possibilità di raccogliere dati e statistiche che permettono di effettuare un'analisi complessiva delle performance del sistema con diverse configurazioni.

6.1 Configurazioni simulate

In particolare, si è deciso di eseguire la simulazione su 12 diverse configurazioni in modo da poter andare ad analizzare i risultati al variare del comportamento e del numero di AGV. Le configurazioni eseguite sono state:

- **BT1 A3:** Behaviour type 1 - Number of AGV 3
- **BT1 A6:** Behaviour type 1 - Number of AGV 6
- **BT1 A9:** Behaviour type 1 - Number of AGV 9
- **BT1 A12:** Behaviour type 1 - Number of AGV 12
- **BT2 A3:** Behaviour type 2 - Number of AGV 3
- **BT2 A6:** Behaviour type 2 - Number of AGV 6
- **BT2 A9:** Behaviour type 2 - Number of AGV 9
- **BT2 A12:** Behaviour type 2 - Number of AGV 12
- **BT3 A3:** Behaviour type 3 - Number of AGV 3
- **BT3 A6:** Behaviour type 3 - Number of AGV 6
- **BT3 A9:** Behaviour type 3 - Number of AGV 9
- **BT3 A12:** Behaviour type 3 - Number of AGV 12

6.2 Lista degli ordini per simulazione

La lista di ordini considerata per effettuare le diverse simulazioni del sistema è stata scelta in maniera semi arbitraria. Come prima cosa è stato indispensabile aggregare articoli appartenente a classi diverse nella stessa classe oppure al contrario dividere una singola classe di articoli in più sottoclassi. Tutto ciò ha permesso di ottenere un numero di classi adeguato alla simulazione sul sistema implementato.

Successivamente si sono selezionati N ordini casuali da quelli disponibili, mantenendo invariate le distribuzioni di classi degli articoli all'interno di essi.

E' possibile notare come gli ordini che indirizzati al cliente "CO" sono di gran lunga più numerosi rispetto a quelli indirizzati a "FI" e "MI": questo rispecchia la distribuzione di ordini suddivisi per cliente presente all'interno degli ordini realmente raccolti dalla logistica.

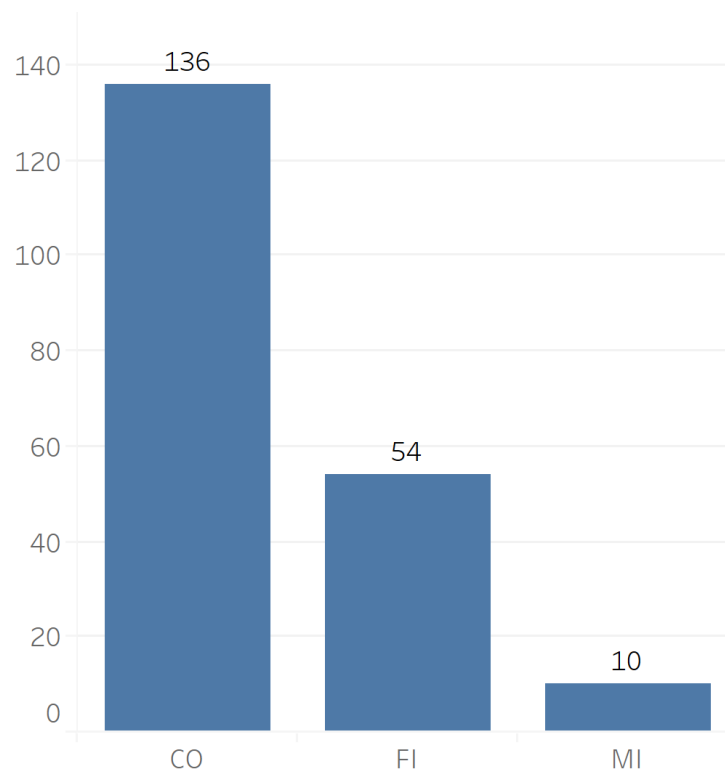


Figure 14: Numero di ordini associati ad ogni cliente.

Inoltre è possibile vedere dal seguente grafico come alcuni articoli compaiono molto più spesso all'interno di determinati clienti rispetto ad altri. Tutto ciò è stato mantenuto di proposito all'interno della lista degli ordini utilizzata per le simulazioni in modo tale da renderla il più verosimile possibile e andare a considerare tutte quelle esigenze realmente presenti all'interno del contesto analizzato.

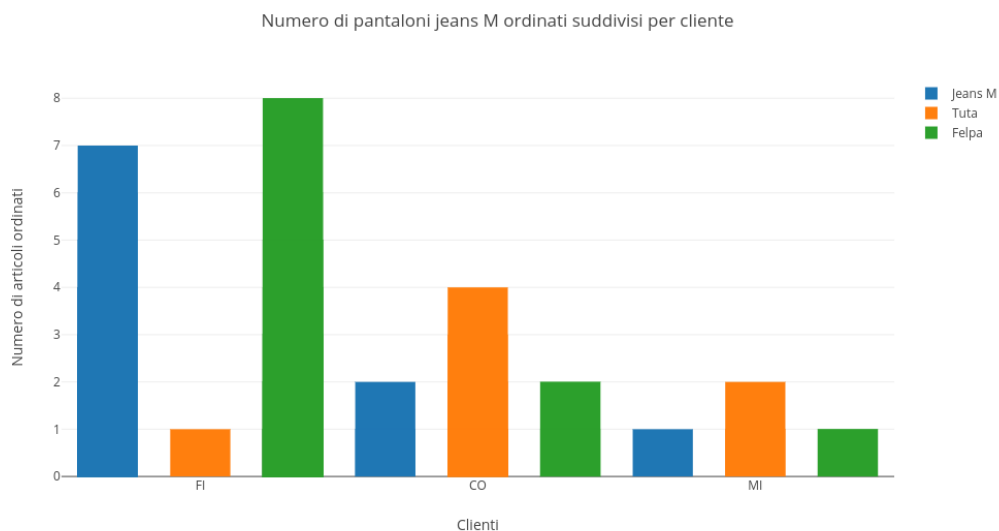


Figure 15: Numero di diversi articoli ordinati per cliente

6.3 Metriche di valutazione

Per poter valutare le performance delle configurazioni simulate è stato realizzato un sistema di raccolta dati che ha permesso di raccogliere diverse informazioni, tra cui:

- **Conflicts:** numero di conflitti che sono stati rilevati durante la configurazione per ogni singolo AGV.
- **Conflict Wait:** numero di conflitti che sono stati rilevati durante la configurazione per ogni singolo AGV e che sono stati risolti con un AGV in attesa per un turno.
- **Conflict Path:** numero di conflitti che sono stati rilevati durante la configurazione per ogni singolo AGV e che sono stati risolti con un AGV che ricalcola il percorso per la sua destinazione.
- **Waiting Gate:** numero di timestep trascorsi da ciascun AGV all'interno delle aree di attesa.
- **Articles:** numero di articoli elaborati, e quindi portati ad uno specifico gate, da ciascun AGV.
- **Moving Steps:** numero di step, ovvero di spostamenti da una cella ad un'altra adiacente, di ciascun AGV.

Questi parametri sono stati raccolti ad ogni timestep della simulazione per ogni singolo AGV per ogni configurazione, così da poter analizzare i risultati che si otterranno secondo diversi punti di vista.

7 Analisi dei risultati

Una volta raccolti i dati inerenti alle diverse simulazioni, si è passati ad analizzarli, con lo scopo di determinare la configurazione migliore: prima di procedere è indispensabile però dare una definizione di "configurazione migliore".

Sicuramente una configurazione può essere considerata più performante di un'altra se impiega meno tempo per portare a termine la stessa mole di lavoro. Andando però a considerarlo solo il tempo impegnato si trascurano alcuni aspetti che sono indispensabili per un'attività reale come in questo caso. E' infatti indispensabile trovare anche un giusto compromesso sul numero di AGV, dato che rappresentano un costo non indifferente di acquisto e mantenimento da parte delle aziende.

Lo scopo di questa fase quindi è proprio quello di individuare come i parametri delle configurazioni influiscono sulle performance del sistema, in modo tale da poter trovare la configurazione migliore tenendo ben presente il contesto lavorativo ed economico in cui questo lavoro si colloca.

In questa sezione sono riportati, descritti e analizzati alcuni grafici rappresentativi dei dati raccolti durante la fase di simulazione e che sono stati considerati più rilevanti. In particolare si andrà ad analizzare:

- Timestep per ciascuna configurazione
- Conflitti per ciascuna configurazione
- Articoli elaborati da ciascun AGV
- Timestep di attesa per ciascun AGV

7.1 Analisi generale delle configurazioni

7.1.1 Timestep per ciascuna configurazione

Come prima cosa si è andati ad analizzare il tempo richiesto da ciascuna configurazione per elaborare la lista di ordini simulata, essendo questa una metrica di valutazione fondamentale per una possibile implementazione reale del sistema. Il seguente grafico riporta il tempo totale richiesto da ciascuna delle 12 diverse simulazioni avviate.

Andando ad analizzare le configurazioni testate con 3 AGV si può vedere come i tempi richiesti dalle tre diverse simulazioni non presentano grandi differenze tra loro. E' possibile notare, pero, come all'aumentare del numero di AGV questa differenza aumenta significativamente, con le configurazioni testate con BT3 che presentano tempi totali largamente inferiori rispetto alle altre configurazioni testate.

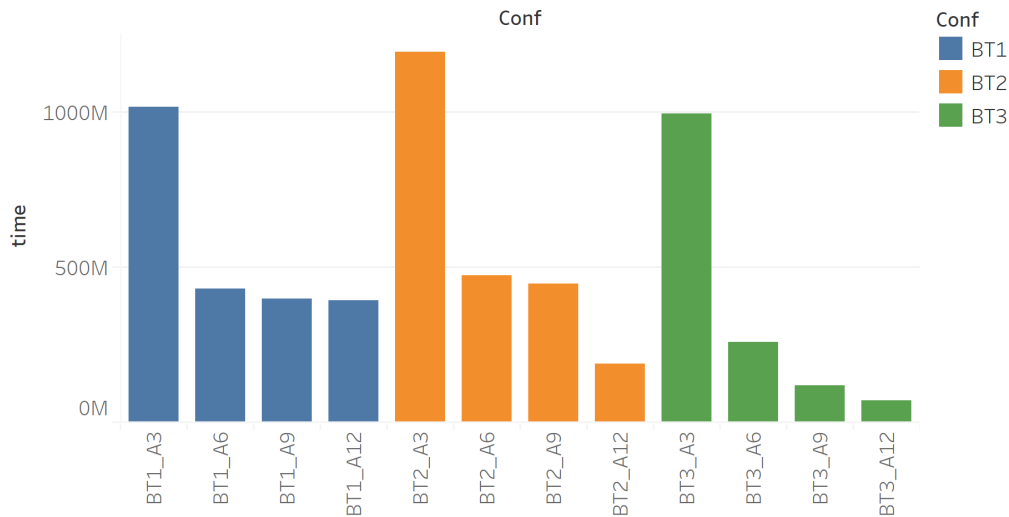


Figure 16: Tempo necessario al completamento degli ordini per ciascuna configurazione

7.1.2 Conflitti per ciascuna configurazione

Un'altra metrica che deve essere presa sicuramente in considerazione per eventuali decisioni future di realizzazione del sistema è il numero totale di conflitti tra AGV successi durante il ciclo di vita di una simulazione.

Il seguente grafico riporta sull'asse delle ascisse il tempo richiesto dalla simulazione per terminare gli ordini e sull'asse delle ordinate il numero totale di conflitti. Si può andare ad identificare un forte rapporto inversamente proporzionale tra tempo richiesto al completo degli ordini in una certa configurazione e il relativo numero di conflitti avvenuti. Infatti, al diminuire del tempo richiesto si presenta un forte aumento del numero di conflitti. Questo rapporto è estremamente significativo su configurazioni con un alto numero di AGV, infatti il seguente grafico riporta le tre configurazioni testate con 12 agenti.

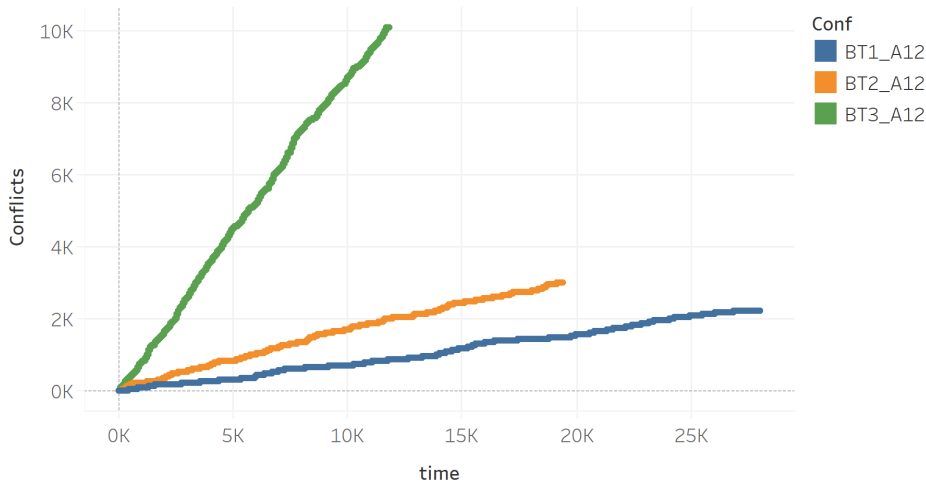


Figure 17: Rapporto tra tempo richiesto e conflitti avvenuti in ciascuna configurazione

7.2 Analisi singoli AGV delle configurazioni

7.2.1 Articoli elaborati da ciascun AGV

I grafici sotto illustrati riportano il numero di articoli correttamente elaborati da ogni singolo AGV per ciascuna configurazione durante l'esecuzione della simulazione. Questa metrica è di estrema importanza in quanto permette di verificare se esistono differenze significative tra la mole di lavoro assegnata ad un AGV rispetto ad un altro all'interno della stessa simulazione.

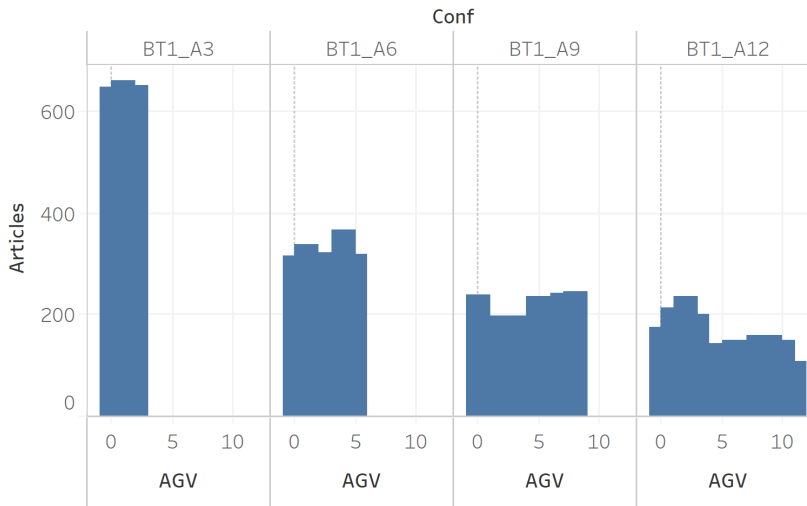


Figure 18: Numero di articoli elaborati da ciascun AGV con BT1

La BT1 [Fig 18] mantiene un carico di lavoro bilanciato quando la simulazione è eseguita con un numero ridotto di AGV, per poi andare a perdere questa distribuzione equa nelle configurazioni con numerosi AGV.

Invece, come si può vedere di seguito, la configurazione BT2 [Fig 19] presenta delle differenze significative per qualsiasi numero di AGV, mostrando degli agenti che, durante un singolo ciclo di simulazione, elaborano anche il doppio degli articoli elaborati da altri agenti nella stessa simulazione.

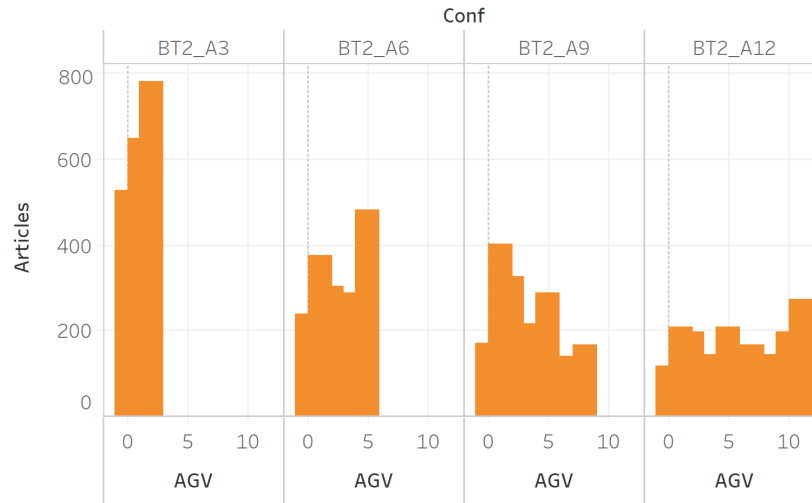


Figure 19: Numero di articoli elaborati da ciascun AGV con BT2

Per quanto riguarda invece la BT3 [Fig 20] si è rilevata di gran lunga la configurazione che meglio distribuisce la mole di lavoro tra gli AGV, indipendentemente dal numero di AGV operativi. Ciò permette non solo di ridurre il tempo di elaborazione totale degli ordini, ma anche di non sovraccaricare di lavoro un determinato agente che potrebbe danneggiarsi o richiedere manutenzione preventiva, che implicherebbero soldi e tempo.

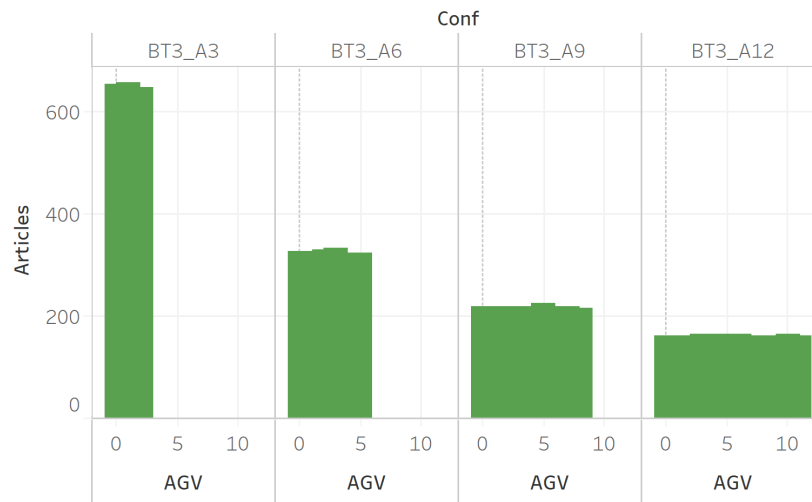


Figure 20: Numero di articoli elaborati da ciascun AGV con BT3

7.2.2 Timestep di attesa per ciascun AGV

Un'altra metrica che ci ha permesso di raggiungere l'obiettivo di questo lavoro è il numero di step temporali passati in attesa da ciascun AGV di ogni simulazione avviata. Considerando il dominio su cui questo lavoro è stato svolto, una delle priorità maggiori è proprio quella di ottimizzare l'utilizzo delle risorse, andando quindi a minimizzare quei momenti in cui le risorse non sono operative.

Il seguente grafico riporta il numero di step temporali attesi da ciascun AGV nelle tre diverse BT durante la simulazione con 12 AGV.

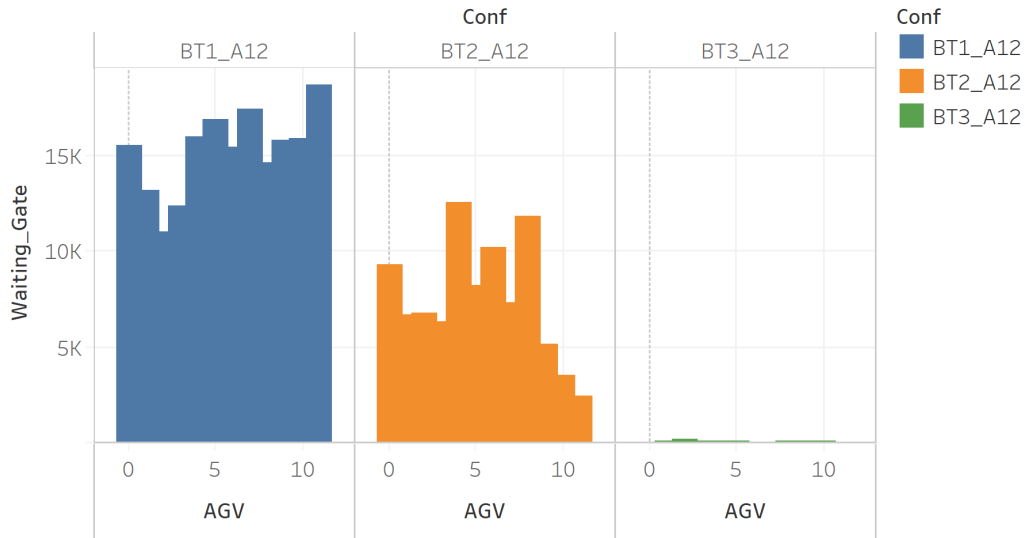


Figure 21: Numero di step temporali passati in attesa da ciascun AGV per i tre diversi BT

Come risulta visibile dal grafico sopra riportato la differenza tra il numero di step attesi da ciascun AGV durante le prime due configurazioni comparate con la BT3 è enorme. La BT3 infatti tende ad azzerare il numero di step spesi in attesa da ciascun AGV.

Come riportato nel seguente grafico, questa differenza diminuisce al diminuire del numero di AGV nelle simulazioni, andando sempre però a riportare una significativa differenza tra le prime due e la terza configurazione.

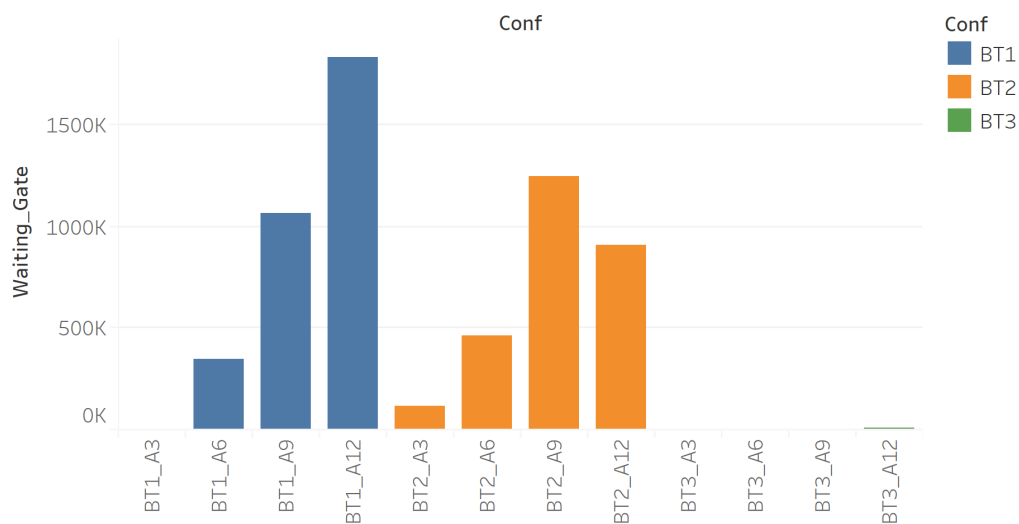


Figure 22: Numero totale di step passati in attesa dagli AGV di una configurazione

8 Conclusioni

- BT3 presenta dei tempi di elaborazione decisamente più bassi di BT1 e BT2, in particolare all'aumentare del numero di AGV disponibili.
- BT3 seppur presenta una considerevole diminuzione del tempo di elaborazione degli ordini, presenta un elevato numero di conflitti.
- BT3 distribuisce il lavoro a ciascun AGV indipendentemente dalla loro numerosità, mentre BT1 solo su pochi AGV e invece BT2 non lo fa proprio.
- BT3 tende ad azzerare i tempi in cui delle risorse siano in attesa, sia con pochi sia con tanti AGV. BT1 e BT2 invece presentano entrambi dei sostanziali tempi di attesa per tutti gli AGV durante le simulazioni. Al diminuir del numero di AGV diminuisce anche questa differenza, seppur rimanendo sempre presente (BT2 leggermente meglio di BT1).

BT3 è la migliore, indipendentemente dal numero di AGV. Considerare che, per ottenere delle performance di gran lunga migliore delle altre configurazioni, bisogna investire per un elevato numero di AGV. Notare che, come detto prima, se si decide di utilizzare un elevato numero di AGV, quindi prestare molta più attenzione alla gestione di conflitti e cercare di ottimizzarla.

Si potrebbero preferire il BT1 e il BT2 solo in casi particolari:

La BT1 ...

La BT2 potrebbe essere di estrema utilità nel caso in cui ci siano diversi tipi di AGV che si occupano di articoli diversi per questione prettamente fisiche/logistiche. In quel caso allora si potrebbe preferire una soluzione di questo genere, che assegna a ciascun AGV un set limitato di tipi di articoli prelevabili.

8.1 Lavori e sviluppi futuri

I risultati ottenuti da questo lavoro sono stati soddisfacenti: nonostante ciò si potrebbe portare avanti questa ricerca, ampliandola e migliorandola sotto diversi aspetti come:

- Gestione della batteria degli AGV:
- Gestione di danni o rotture degli AGV:
- Gestione di sistema con AGV eterogene:
- Gestione navigazione degli AGV: algoritmo A* oppure navigazione su corsie.

References

- [1] *Tableau software*, 2013.
- [2] Python Software Foundation, *Python 3.6.7*, 2008.
- [3] Inc. GitHub, *Open source survey*, 2017.
- [4] Massimo Cossentino Carmelo Lodato Salvatore Lopes Patrizia Ribino, *Multi agent simulation for decision making in warehouse management.*, (2006)., ISBN 978-83-60810-39-2978-83-.
- [5] Yael Edan Sigal Berman, Edna Schechtman, *Evaluation of automatic guided vehicle systems.*, (2006)., Contents lists available at ScienceDirect.
- [6] Wikipedia, *Sistema complesso — wikipedia, l'enciclopedia libera*, 2018.
- [7] ———, *Sistema multiagente — wikipedia, l'enciclopedia libera*, 2018.
- [8] ———, *Agente intelligente — wikipedia, l'enciclopedia libera*, 2019.
- [9] Wikipedia contributors, *Lee algorithm — Wikipedia, the free encyclopedia*, 2018, [Online; accessed 4-April-2019].
- [10] ———, *Breadth-first search — Wikipedia, the free encyclopedia*, 2019, [Online; accessed 4-April-2019].
- [11] Chun Wong, *Pycxsimulator*, 2012.
- [12] Michael Wooldridge, *An introduction to multiagent systems.*, (2002)., ISBN 978-0-471-49691-5.