UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione



TESI DI LAUREA

Pianificazione della consegna dell'ultimo miglio basata su ipergrafi tipizzati con orchestrazione semantica tramite LLM

Last-mile delivery planning based on typer hypergraphs with LLM semantic orchestration

Relatore

Dott. Enrico Corradini

Candidato

Mattia Mandorlini

ANNO ACCADEMICO 2025-2026

A chiunque abbia condiviso con me un pezzo di questo cammino, regalandomi un sorriso, un consiglio o un momento di ascolto.

Sommario

La crescita esponenziale dell'e-commerce e dei servizi di consegna rapida ha reso la logistica dell'ultimo miglio (*last-mile delivery*) una delle sfide operative più rilevanti e complesse per la mobilità urbana contemporanea. Gestire flotte di corrieri sul territorio richiede di armonizzare molteplici esigenze: garantire consegne tempestive entro finestre orarie concordate, ridurre l'impatto del traffico e assicurare carichi di lavoro equi e sostenibili per gli operatori. Tuttavia, mentre le tecniche classiche di ottimizzazione matematica faticano ad adattarsi a imprevisti e a istruzioni non strutturate, i modelli di Intelligenza Artificiale generativa pura offrono grandi capacità interpretative ma difettano di comprensione della topologia della rete e rispetto rigoroso dei vincoli fisici. L'obiettivo di questa tesi è superare tale dicotomia progettando un'architettura ibrida che unisce la modellazione delle reti complesse tramite ipergrafi tipizzati con l'orchestrazione agentica guidata da modelli linguistici di grandi dimensioni (*Large Language Models*). Il lavoro definisce un approccio articolato su due orizzonti decisionali, integrando la pianificazione strategica mattutina con la gestione reattiva delle richieste pomeridiane in streaming. Il sistema viene infine validato su dati operativi reali, dimostrando la capacità di migliorare la puntualità del servizio, contenere le percorrenze complessive, equilibrare il carico tra gli operatori e garantire risposte stabili ed efficienti nell'operatività quotidiana.

Keyword: Last-Mile delivery, Logistica urbana, Intelligenza Artificiale, Ipergrafi, Large Language Models (LLM), Sistemi agentici.

Indice

Indice ii

Introduzione 1

1 Background teorico e stato dell'arte 3
1.1 Logistica dell'ultimo miglio e problemi di routing . . . 3
1.1.1 Vehicle Routing Problem with Time Windows . . . 3
1.1.2 Programmazione Matematica . . . 4
1.2 Teoria dei grafi e Reti complesse . . . 5
1.2.1 Ipergrafi . . . 5
1.2.2 Community Detection . . . 7
1.2.3 Algoritmi di diffusione . . . 8
1.2.4 Personalized PageRank . . . 8
1.2.5 Centralità Topologiche . . . 8
1.3 Large Language Models . . . 9
1.3.1 Chain-of-Thought . . . 9
1.3.2 Sistemi Agentici . . . 10

2 Ipergrafo tipizzato e orchestrazione agentica 13
2.1 Architettura generale e orizzonti decisionali . . . 13
2.1.1 Limiti delle rappresentazioni tradizionali a grafo . . . 14
2.1.2 L'ipergrafo tipizzato . . . 14
2.2 Costruzione dell'ipergrafo tipizzato . . . 14
2.2.1 Vertici, iperarchi e matrice di incidenza . . . 15
2.2.2 Tassonomia matematica dei cinque iperarchi . . . 15
2.2.3 Commensurabilità dei pesi e gestione della sparsità . . . 18
2.3 Grafi derivati e transizioni markoviane . . . 18
2.3.1 Grafi ausiliari e proiezioni tipizzate . . . 18
2.3.2 Kernel di cammino casuale Vertex–Hyperedge–Vertex (V-E-V) . . . 19
2.3.3 Formulazione matriciale . . . 20
2.4 Le metriche principali . . . 20
2.4.1 Misure di affinità e compatibilità . . . 20
2.4.2 Partizionamento spaziotemporale e PDCC-H . . . 21
2.4.3 Text-Informed Centrality (TIMC) . . . 22
2.5 Politiche Decisionali . . . 22

2.5.1 Assegnazione delle comunità e bilanciamento dei carichi 22
2.5.2 Pianificazione della rotta locale 23
2.5.3 Inserimento dinamico in streaming 23
2.6 Controllo agentico e guardrail di sicurezza 24
2.6.1 Ruolo dell'Orchestratore LLM 24
2.6.2 La pipeline di sicurezza 25
2.6.3 Formulazione matematica del vincolo di Trust-Region 25

3 Validazione sperimentale e analisi delle prestazioni 26
3.1 Setup sperimentale e protocollo di valutazione 26
3.1.1 Dataset e generatore sintetico 26
3.1.2 Indicatori di prestazione e tracciabilità agentica 27
3.1.3 Protocollo di validazione statistica a blocchi 29
3.2 Prestazioni e meccanismi operativi del sistema 30
3.2.1 Risultati della pianificazione mattutina 30
3.2.2 Risultati del matching pomeridiano in streaming 31
3.2.3 Valutazione dell'orchestratore LLM 31
3.3 Studio di ablazione 32
3.3.1 Contributo causale delle metriche sull'ipergrafo 33
3.3.2 Valutazione della centralità TIMC 33
3.4 Analisi di sensibilità agli iperparametri e robustezza 34
3.4.1 Sensibilità alla risoluzione delle comunità 34
3.4.2 Sensibilità al restart di PageRank e alla regione di fiducia 34
3.5 Scalabilità computazionale e latenza operativa 35
3.5.1 Complessità algebrica e aggiornamenti locali sparsi 35
3.5.2 Distribuzione delle latenze di inserimento in streaming 36
3.6 Studi di caso qualitativi 36
3.6.1 Scenario A: Centro urbano ad alta densità 36
3.6.2 Scenario B: Periferia a bassa densità 36

4 Valutazione comparativa con le metodologie di riferimento 38
4.1 Definizione delle metodologie di confronto 38
4.1.1 Nearest Neighbor (NN) 38
4.1.2 Proiezione su grafi tradizionali a coppie: Projection-Only 39
4.1.3 Solutori esatti 39
4.1.4 Modelli generativi non vincolati 39
4.1.5 Quadro sinottico delle proprietà architetturali 40
4.2 Risultati sperimentali comparativi 40
4.2.1 Analisi complessiva delle prestazioni 40
4.2.2 Efficienza chilometrica e impatto ambientale 41
4.2.3 Puntualità delle consegne 41
4.2.4 Equità del carico di lavoro tra i corrieri 42
4.2.5 Validazione statistica su benchmark multi-finestra 42
4.3 L'impatto dell'orchestratore agentico sui vincoli operativi testuali 43
4.3.1 La cecità dei metodi tradizionali verso le note non strutturate 43
4.4 Studio di un caso d'uso operativo 44
4.4.1 Descrizione dello scenario e arrivo dell'evento dinamico 44
4.4.2 Confronto delle risposte operative generate dai modelli 44

5 Discussione e conclusioni **46**
5.1 Discussione 46
5.1.1 Riflessioni operative sui paradigmi a confronto 46
5.1.2 Integrazione tra modelli matematici e agenti AI 47
5.1.3 Impatto umano, sociale e sostenibilità urbana 47
5.1.4 Contesti di massima efficacia e limiti metodologici dell'indagine . . . 47
5.1.5 Sviluppi futuri 48
5.2 Conclusioni 49

Bibliografia **51**

Elenco delle figure

1.1 Esempio di VRPTW 4
1.2 Confronto tra grafo e ipergrafo 6
1.3 Architettura dei sistemi agentici 11

4.1 Confronto visivo delle metriche operative primarie tra la Pipeline a Ipergrafo, l'euristica Nearest Neighbor e il modello Projection-Only sul dataset reale di Jilin. 41

Elenco delle tabelle

3.1 Prestazioni medie sui 5 giorni di test nelle quattro condizioni operative, in regime nominale e sotto disruzione di traffico (shock). 32

4.1 Quadro comparativo delle caratteristiche architetturali tra i modelli di instradamento. 40
4.2 Confronto delle prestazioni primarie sul test set reale di Jilin (250 ritiri pomeridiani in streaming). 41
4.3 Miglioramento relativo percentuale ($\Delta\%$) del framework a ipergrafo rispetto alle metodologie di riferimento e significatività statistica con test di Wilcoxon su benchmark multi-finestra. 43

Introduzione

La logistica dell'ultimo miglio (last-mile delivery), ovvero la fase terminale della catena di fornitura in cui le merci vengono trasferite dai centri di smistamento periferici fino alla destinazione finale, rappresenta uno dei segmenti più critici, complessi e dinamici dell'economia contemporanea. La crescita esponenziale dell'e-commerce, accelerata dai mutamenti nelle abitudini di consumo e dalla crescente richiesta di servizi di consegna rapida e personalizzata in giornata, ha trasformato radicalmente il panorama della mobilità e della distribuzione urbana. Tuttavia, questa espansione porta con sé sfide logistiche e sociali di primaria grandezza: l'intensificazione del traffico stradale, l'aumento delle emissioni inquinanti e di particolato nelle aree metropolitane ad alta densità abitativa, la necessità di rispettare rigide finestre orarie concordate con i clienti e l'esigenza impellente di garantire carichi di lavoro equi e sostenibili per gli operatori sul territorio. In questo scenario articolato, l'ottimizzazione dei percorsi e del dispacciamento della flotta non costituisce soltanto una leva di efficienza economica per le aziende del settore, ma diventa un elemento determinante per la transizione ecologica e la vivibilità delle città moderne.

La trattazione proposta nasce dal profondo interesse nell'esplorare la convergenza tra due frontiere scientifiche tradizionalmente distinte ma straordinariamente complementari: la matematica discreta applicata all'analisi delle reti complesse ad alto ordine e l'Intelligenza Artificiale basata su modelli linguistici di grandi dimensioni (*Large Language Models*, LLM) operanti all'interno di architetture agentiche autonome. Da un punto di vista scientifico e accademico, la logistica dell'ultimo miglio offre un banco di prova ideale per verificare se e come le emergenti capacità di comprensione semantica e ragionamento strategico degli LLM possano integrarsi armoniosamente con modelli algebrici rigorosi e deterministici. Dal punto di vista applicativo e industriale, vi è la forte motivazione di progettare una soluzione concreta, trasparente e scalabile, capace di superare tanto i limiti computazionali delle metodologie tradizionali quanto i rischi di inaffidabilità dei moderni sistemi di intelligenza artificiale generativa pura.

Nell'affrontare la pianificazione logistica metropolitana, la letteratura scientifica e la pratica industriale si scontrano infatti con due ostacoli strutturali: da un lato, i solutori esatti della Ricerca Operativa classica affrontano una complessità combinatoria esponenziale che ne impedisce l'impiego per la gestione di flussi dinamici pomeridiani in tempo reale, mentre i grafi a coppie tradizionali comprimono le relazioni multi-entità tra corrieri, aree territoriali, depositi e compiti di consegna disperdendo preziose informazioni topologiche; dall'altro, i tentativi di delegare l'instradamento direttamente a modelli linguistici generativi falliscono a causa di allucinazioni toponomastiche, lentezza di risposta e incapacità di rispettare rigorosamente i vincoli geometrici e le capacità dei veicoli. Tuttavia, i modelli linguistici possiedono

l'abilità unica di comprendere le istruzioni espresse in testo libero dai clienti, informazione essenziale che i solutori matematici convenzionali ignorano sistematicamente.

Per superare tale dicotomia, il presente lavoro sviluppa e formalizza un'architettura ibrida fondata su un ipergrafo tipizzato e pesato, articolata su due orizzonti decisionali cooperanti. Durante la pianificazione mattutina, le relazioni poliadiche del sistema vengono modellate senza perdita informativa su ipergrafo, impiegando processi markoviani di diffusione con riavvio per stimare l'affinità territoriale tra corrieri e zone, e raggruppando le consegne in comunità di lavoro stabili mediante l'algoritmo di Leiden affiancato da euristiche locali di instradamento. Nel corso del pomeriggio, un motore dinamico locale processa reattivamente le nuove richieste di ritiro in streaming, mentre a livello macroscopico un Orchestratore Agentico basato su LLM interpreta le note operative in linguaggio naturale e modula i parametri strategici della flotta. L'affidabilità del sistema è garantita da una pipeline di guardrail deterministici a sicurezza geometrica, che impedisce deviazioni operative e allucinazioni decisionali.

L'intero impianto metodologico è stato validato empiricamente sulle tracce industriali reali della città di Jilin tratte dal dataset Cainiao LaDe, operando su una complessa rete stradale metropolitana e su flussi concorrenti di ordini dinamici. I risultati evidenziano come l'approccio proposto riesca a conciliare un elevato tasso di puntualità con un drastico abbattimento delle percorrenze chilometriche complessive, assicurando al contempo una ripartizione equa e sostenibile dei carichi di lavoro tra gli operatori e una sensibile riduzione dei compiti rifiutati sul campo grazie all'interpretazione delle note di consegna.

La presente tesi è strutturata in quattro capitoli principali seguiti da un capitolo di sintesi conclusiva, articolati come segue: nel Capitolo 1 si esaminano i fondamenti teorici e lo stato dell'arte multidisciplinare relativi ai problemi di instradamento veicolare con finestre temporali, alla teoria degli ipergrafi e ai modelli linguistici all'interno di architetture agentiche; nel Capitolo 2 si formalizza l'architettura ibrida proposta, descrivendo la costruzione dell'ipergrafo tipizzato, le metriche di diffusione markoviana, le politiche decisionali a doppio orizzonte e la pipeline di sicurezza; nel Capitolo 3 si sviluppa la validazione sperimentale interna e l'analisi di sensibilità e ablazione sulle tracce industriali della città di Jilin, verificando la significatività statistica e la scalabilità computazionale della soluzione; nel Capitolo 4 si conduce una dettagliata valutazione comparativa tra il modello proposto e le principali metodologie di riferimento della letteratura scientifica e industriale, approfondendo i compromessi operativi ed esaminando un caso d'uso applicativo sul campo; infine, nel capitolo conclusivo si discutono i risultati ottenuti, i limiti metodologici emersi e le direttrici più promettenti per i futuri sviluppi della ricerca.

CAPITOLO 1

Background teorico e stato dell'arte

In questo capitolo vengono presentati i fondamenti teorici e lo stato dell'arte alla base del modello proposto. La trattazione introduce i problemi di instradamento dell'ultimo miglio (VRP e VRPTW), per poi esaminare la teoria dei grafi complessi e degli iperarchi con i relativi processi di diffusione e community detection. Infine, viene approfondita l'architettura dei Large Language Models (LLM) all'interno dei sistemi agentici e dei guardrail di sicurezza.

1.1 Logistica dell'ultimo miglio e problemi di routing

Il Vehicle Routing Problem (VRP) [Toth e Vigo, 2014] è un problema di ottimizzazione che mira a trovare le rotte ottimali per una flotta di veicoli incaricata di visitare un insieme di posizioni. Il VRP rappresenta la generalizzazione del noto problema del commesso viaggiatore estesa a veicoli multipli. L'obiettivo per una rotta "ottimale" spesso non è solo percorrere la distanza totale minima, ma anche minimizzare la lunghezza della singola rotta più lunga, al fine di completare tutte le consegne nel minor tempo possibile. Questi problemi sono intrinsecamente intrattabili: all'aumentare delle posizioni, il numero delle rotte possibili esplode in maniera esponenziale. Ad esempio, con soli 20 punti da visitare, il numero di permutazioni possibili supera i trilioni, rendendo impraticabile una ricerca esaustiva dello spazio delle soluzioni.

1.1.1 Vehicle Routing Problem with Time Windows

Il Vehicle Routing Problem with Time Windows (VRPTW) [Solomon, 1987] è una generalizzazione del VRP in cui i veicoli devono servire i clienti rispettando specifici intervalli temporali, definiti come finestre di tempo (time windows). A causa dei rigidi vincoli di orario, la pianificazione necessita di una matrice dei tempi di percorrenza, anziché di una semplice matrice delle distanze, in modo da incorporare le dipendenze temporali. Il modello deve gestire simultaneamente sia i limiti di tempo, sia i vincoli di capacità massima del veicolo legati a peso o volume del carico. La risoluzione pratica determina una "finestra di soluzione", che rappresenta l'intervallo esatto in cui un veicolo può arrivare senza violare i tempi della rotta generale. In caso di arrivo anticipato del veicolo, il modello di ottimizzazione gestisce attivamente i tempi di attesa (slack) affinché la consegna rispetti l'orario di apertura della finestra temporale.

Il diagramma in Figura 1.1 mostra le località da visitare in blu e il deposito in nero. Le finestre temporali sono visualizzate sopra ogni località. L'obiettivo è ridurre al minimo il tempo di percorrenza totale dei veicoli.

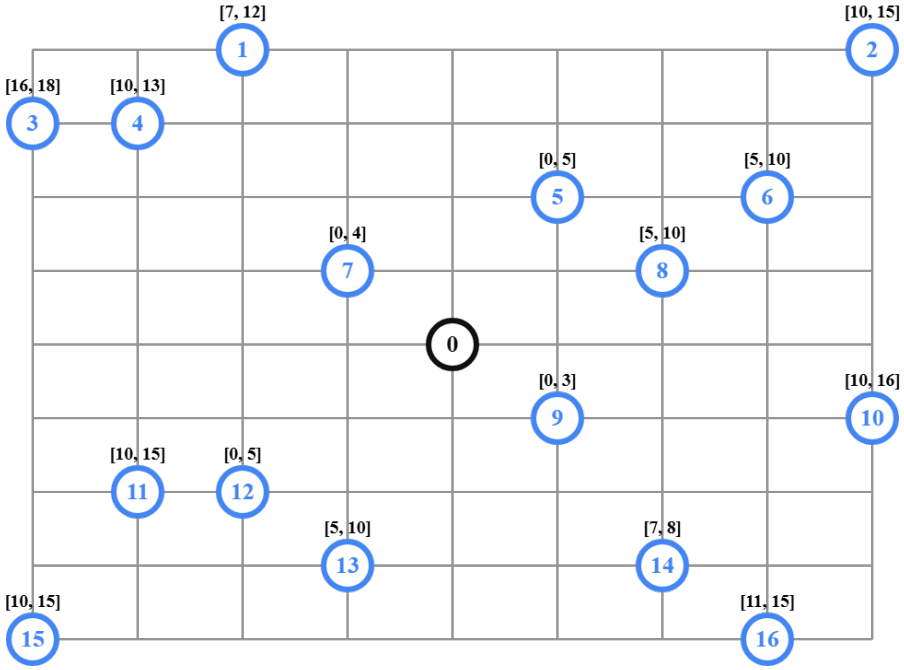


Figura 1.1: Esempio di VRPTW

Streaming Pickups

La logistica urbana deve gestire una tensione strutturale tra due fasi: la pianificazione mattutina statica e il coordinamento dinamico pomeridiano. Nella logistica statica (ad orizzonte fisso), tutte le consegne e le informazioni sono note in anticipo e si pianificano raggruppamenti su veicoli limitati. La logistica dinamica interviene quando nuovi prelievi (streaming pickups) o richieste dell'ultimo minuto si generano mentre i rider sono già lungo il tragitto [Frohner e Raidl, 2020]. In ambienti puramente dinamici e stocastici, il sistema non conosce l'esatta conformazione degli ordini futuri e deve operare attraverso un replanning continuo in base al manifestarsi degli eventi. Le soluzioni statiche tradizionali faticano ad adattarsi a questo scenario, motivo per cui la gestione dinamica processa le nuove richieste abbinandole al momento e inserendole miopicamente all'interno della sequenza del rider attivo in base alla capacità residua e alle finestre temporali.

1.1.2 Programmazione Matematica

Il Mixed-Integer Linear Programming (MILP) viene impiegato per formalizzare modelli matematici esatti di ottimizzazione per problemi come le consegne logistiche e il routing. I risolutori basati sulla programmazione matematica si affidano tipicamente a tecniche di ricerca sistematica, come gli algoritmi di Branch-and-Bound (B&B), che valutano i confini matematici e i rilassamenti del problema nel tentativo di garantire teoricamente una soluzione ottimale. Tuttavia, poiché il VRP è un problema a complessità NP-Hard, l'approccio esatto va inevitabilmente incontro all'esplosione combinatoria. La complessità computazionale cresce in maniera esponenziale, tanto che, all'aumentare dei nodi, i solver possono impiegare ore,

giorni o anni per raggiungere la convergenza. Sotto stringenti limiti di tempo reale, il solver va in timeout fornendo soluzioni frammentate e ottimizzate solo a livello locale.

Euristiche e Approcci Solver-Free

Le limitazioni dei solutori esatti in contesti in tempo reale spingono l'industria logistica verso l'uso di euristiche e approcci solver-free. Le euristiche sacrificano la garanzia di ottimalità a favore dell'efficienza computazionale, costruendo soluzioni "sufficientemente buone" entro finestre operative ridottissime (nell'ordine di pochi secondi o millisecondi).

In presenza di ordini stocastici in arrivo, ricalcolare da zero l'intera istanza VRP mediante un solutore esatto risulta impraticabile per via dei tempi di calcolo incompatibili con i vincoli real-time. In questi contesti, euristiche costruttive più rapide (come quelle di tipo greedy) consentono un dispatching immediato: pur a fronte di una soluzione sub-ottimale rispetto a metaeuristiche più onerose (quali Adaptive Large Neighborhood Search o Algoritmi Genetici), abbattono la latenza computazionale di diversi ordini di grandezza.

1.2 Teoria dei grafi e Reti complesse

Nella matematica discreta, un grafo semplice $G = (V, E)$ è definito matematicamente come una coppia composta da un insieme non vuoto di vertici (o nodi) V e da un insieme E contenente coppie di elementi di V, chiamati archi. In questo modello, ogni arco collega esattamente due vertici, rendendolo ideale per rappresentare le relazioni 1-a-1 (o a coppie) tra entità. I grafi semplici non ammettono archi che collegano un nodo a se stesso né archi multipli tra la stessa coppia di vertici.

Una classe particolare è costituita dai grafi bipartiti, in cui l'insieme dei vertici è suddiviso in due partizioni disgiunte ($V1$ e $V2$) in modo tale che ogni arco colleghi esclusivamente un vertice di $V1$ con un vertice di $V2$, senza connessioni interne alle singole partizioni.

Con la crescente necessità di analizzare sistemi reali maggiormente intricati, la teoria classica si è estesa alle reti multilivello (multilayer networks). Queste reti permettono di modellare entità che interagiscono simultaneamente attraverso pattern complicati e molteplici tipologie di connessioni o "aspetti" (ad esempio, lo stesso gruppo di persone connesse tramite relazioni di amicizia, lavorative o su diversi social network). Formalmente, una rete multilivello può essere descritta come un costrutto $M = (V_M, E_M, V, E)$, che estende il tradizionale insieme di nodi e archi introducendo il concetto strutturato di "livello" [Kivelä *et al.*, 2014]. In questo modello matematico, V ed E costituiscono gli insiemi base contenenti rispettivamente i nodi e gli archi dell'intero sistema. La struttura si articola ulteriormente attraverso la componente V_M, che elenca le specifiche combinazioni in cui un determinato nodo è effettivamente presente all'interno di un determinato livello, e l'insieme degli archi multilivello E_M, il quale rappresenta strutturalmente la distinzione tra archi intra-livello e archi inter-livello.

1.2.1 Ipergrafi

Poiché in molti sistemi complessi (reti biologiche, sociali, database) le interazioni coinvolgono più di due entità contemporaneamente, le tradizionali relazioni a coppie dei grafi risultano limitanti.

Un ipergrafo $H = (V, E)$ risolve questo limite fungendo da generalizzazione del grafo standard. Esso è composto da un insieme finito di vertici V e da una famiglia E di sottoinsiemi di V, chiamati iperarchi [Bretto, 2013]. A differenza degli archi normali, un iperarco può contenere o connettere un numero arbitrario di vertici, permettendo una formulazione

matematica robusta e naturale per le relazioni n-arie (molti-a-molti). Una rappresentazione fondamentale di H avviene tramite la sua matrice di incidenza X. In tale matrice, le righe corrispondono ai vertici v_i e le colonne agli iperarchi e_j e i valori seguono la regola $X_{i,j} = 1$ se il vertice $v_i \in e_j$ e 0 altrimenti.

Nella Figura 1.2 è illustrato un confronto tra un grafo standard e un ipergrafo.

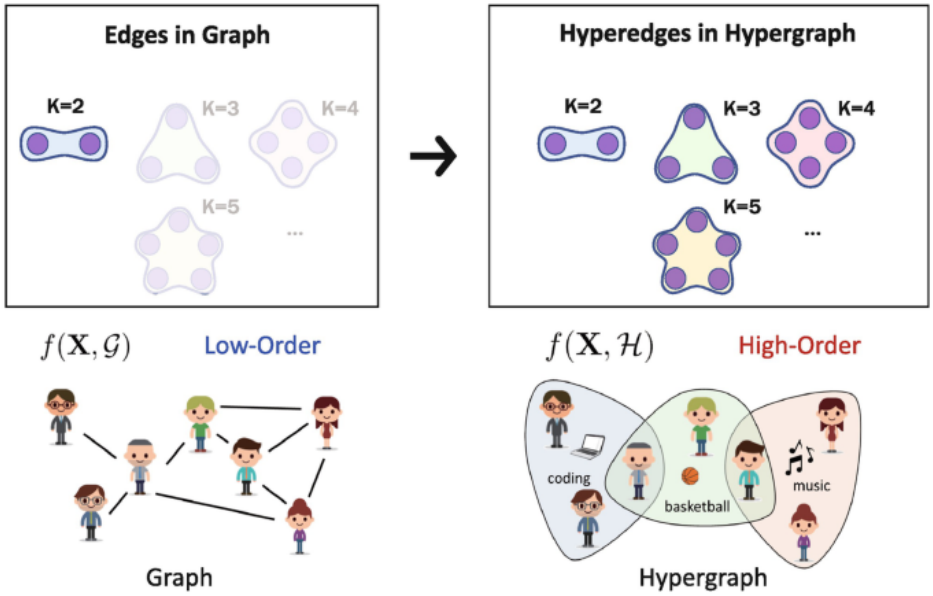


Figura 1.2: Confronto tra grafo e ipergrafo

Grafi duali e Proiezioni

Un ipergrafo può essere manipolato o mappato in altre strutture per applicare algoritmi dell'analisi classica delle reti. Introduciamo tre concetti fondamentali.

Grafo Duale (H^*) Il duale di un ipergrafo capovolge il ruolo di nodi e iperarchi. I vertici del nuovo ipergrafo H^* corrispondono agli iperarchi dell'ipergrafo originale, mentre i nuovi iperarchi sono composti dai vertici originali raggruppati in base agli iperarchi a cui appartenevano. Questa dualità permette di guardare lo stesso sistema da una prospettiva speculare, particolarmente utile nel design dei database.

Line Graph (Grafo degli iperarchi) Il line graph $L(\mathcal{E})$ è un grafo semplice in cui i nodi rappresentano gli iperarchi dell'ipergrafo originale, e un arco collega due nodi se i rispettivi iperarchi originari condividono almeno un vertice in comune. Una generalizzazione di questo concetto sono gli s-line graphs, dove i nodi si collegano solo se condividono almeno s vertici. In contesti in cui gli iperarchi hanno pesi associati e dimensioni differenti, la connessione nel line graph viene modellata in forma pesata. Per quantificare la forza del legame tra due iperarchi normalizzandola rispetto alla loro ampiezza complessiva, si adotta una ponderazione basata sull'indice di similarità di Jaccard tra i nodi condivisi:

$$W_{ee'} = w_e \cdot w_{e'} \frac{|e \cap e'|}{|e \cup e'|} \tag{1.2.1}$$

dove w_e e $w_{e'}$ sono i pesi dei rispettivi iperarchi, $|e \cap e'|$ indica la cardinalità della loro intersezione e $|e \cup e'|$ quella della loro unione. In questo modo, a parità di nodi condivisi, la

correlazione viene opportunamente ridimensionata per iperarchi di grandi dimensioni senza penalizzare l'intensità della sovrapposizione effettiva.

Proiezioni bipartite e Clique expansion Qualsiasi ipergrafo può essere trasformato perfettamente in un grafo bipartito creando due tipologie di nodi: una per i vertici originali e una per gli iperarchi, connettendoli se il vertice appartiene all'iperarco. Un'alternativa molto usata per l'analisi a coppie è la Clique Expansion (o 2-section graph), dove l'ipergrafo collassa in un grafo standard unendo in una clique (sottografo completo) tutti i vertici che co-occorrono all'interno di uno stesso iperarco.

1.2.2 Community Detection

Il rilevamento di comunità (Community Detection) consiste nella partizione o scomposizione di una rete complessa in sottografi (comunità) caratterizzati da un'elevata densità di collegamenti interni e una bassa densità di archi esterni. Tale processo permette di far emergere la struttura mesoscopica del sistema.

L'algoritmo Louvain e i suoi limiti Introdotto originariamente da Blondel et al. [Blondel *et al.*, 2008] per ottimizzare la metrica di modularità[1], Louvain è un metodo agglomerativo euristico e avido (greedy) che opera in due fasi iterative: lo spostamento locale dei singoli nodi per massimizzare il punteggio (local-moving) e la successiva aggregazione delle comunità in super-nodi. Sebbene popolare e molto rapido, presenta un difetto strutturale serio, in quanto le comunità risultanti possono risultare internamente poco o per nulla connesse.

L'algoritmo Leiden Per ovviare ai limiti di Louvain, è stato introdotto l'algoritmo Leiden [Traag *et al.*, 2019]. La sua innovazione risiede nell'inserire una fase di raffinamento tra la mossa locale e l'aggregazione. Durante questa fase, i vertici possono esplorare partizioni alternative all'interno dei confini delle comunità scoperte nel passo precedente, impedendo la frammentazione e assicurando matematicamente che tutte le comunità scoperte siano "ben connesse". L'algoritmo di Leiden si rivela spesso anche computazionalmente più veloce ed efficace.

Constant Potts Model Sia Louvain che Leiden possono ottimizzare al posto della classica modularità, che soffre dei cosiddetti limiti di risoluzione per le reti molto vaste, la metrica Constant Potts Model (CPM). La formulazione del CPM valuta la qualità di una partizione attraverso un bilanciamento tra premi e penalità, definendosi come:

$$H = \sum_c \left(e_c - \gamma \binom{n_c}{2} \right)$$

dove e_c indica il numero di archi interni alla comunità c e n_c il numero di nodi che la compongono (con $\binom{n_c}{2}$ che rappresenta tutte le coppie possibili di nodi al suo interno). Il parametro di risoluzione γ definisce una soglia critica di densità: il modello premia l'aggiunta di archi interni reali, ma applica una penalità pari a γ per ogni coppia di nodi all'interno della stessa comunità. Di conseguenza, un raggruppamento risulta vantaggioso per il modello solo se la sua densità di connessioni interne supera la soglia γ.

[1] La Modularità (Q) è la metrica più usata in Network Science per quantificare la "bontà" di una suddivisione in comunità. In parole semplici: misura quanto i nodi all'interno di un gruppo siano più collegati tra loro di quanto lo sarebbero per pura casualità.

Partizionamento e scomposizione Nel contesto del Vehicle Routing, le comunità mesoscopiche identificate tramite algoritmi di clustering (come Leiden) possono superare la capacità fisica o temporale di un singolo veicolo (comunità definita non ammissibile o infeasible). Si rende quindi necessaria una tecnica di decomposizione geometrica per scindere un cluster in due partizioni coese. La scomposizione ottimale si formula come un problema di bisezione del grafo volto a minimizzare la conduttanza del taglio. Per minimizzare questa quantità, si impiega l'euristica di Kernighan-Lin [Kernighan e Lin, 1970], un algoritmo di bisezione locale che scambia iterativamente coppie di nodi tra le due partizioni per massimizzare il guadagno di taglio e minimizzare la conduttanza.

1.2.3 Algoritmi di diffusione

L'analisi della diffusione modella la propagazione di segnali, flussi di dati e stati operativi attraverso i collegamenti di una rete. Modelli classici includono i modelli a cascata e i modelli epidemici (come il SIR), i quali propagano le informazioni in base alla forza delle connessioni con il vicinato [Singh *et al.*, 2024].

1.2.4 Personalized PageRank

Un algoritmo di diffusione di importanza cruciale è il PageRank, progettato originariamente per calcolare un ranking globale di importanza dei nodi web basato sulla struttura dei link. Il modello matematico del PageRank simula un random surfer che alterna la navigazione lungo gli archi della rete a salti stocastici verso pagine arbitrarie, regolati da un vettore di distribuzione $\mathbf{E}$. Questo meccanismo di restart (o teletrasporto) è indispensabile per evitare che la passeggiata casuale rimanga intrappolata in cicli chiusi (rank sinks). Modificando il vettore di restart in modo che non sia uniforme, ma concentrato su uno o pochi nodi specifici, si ottiene il Personalized PageRank (PPR) [Page *et al.*, 1999]. Il PPR forza la diffusione a "ripartire" sempre dalle vicinanze del nodo sorgente, permettendo di calcolare un ranking di importanza personalizzato e focalizzato su una determinata prospettiva. Questa metrica di diffusione con restart viene impiegata, ad esempio, per quantificare l'affinità locale in reti logistiche (Driver-Area Affinity), applicando una probabilità di restart α per confinare l'esplorazione stocastica al vicinato strutturale di interesse.

1.2.5 Centralità Topologiche

La centralità è una misura quantitativa che esprime l'importanza relativo-strutturale di un nodo (o di un arco) all'interno del sistema complessivo. Non esiste una singola definizione universale di "importanza", la centralità dipende dal ruolo specifico che ci interessa valutare.

Betweenness Centrality

La Betweenness Centrality misura quante volte un nodo si trova lungo i cammini minimi che collegano tutte le possibili coppie di nodi della rete. Un nodo con elevata Betweenness agisce da "ponte", da broker informativo o da collo di bottiglia: se venisse rimosso, i flussi di comunicazione tra molti altri nodi verrebbero interrotti o rallentati.

Closeness Centrality

La Closeness Centrality misura quanto un nodo sia "vicino" a tutti gli altri nodi della rete lungo i cammini minimi. Un nodo con un'elevata Closeness può diffondere informazioni (o accedere a risorse) nell'intera rete nel minor tempo medio possibile.

Coupling Centrality

Nei grafi che modellano sistemi operativi (ad esempio nelle reti di dispatching e routing), si definiscono metriche di centralità ibride come la Pickup-Delivery Coupling Centrality. Questo tipo di centralità non valuta solo la topologia statica, ma la fattibilità di accoppiamento dinamico tra due nodi (es. un autista e un nuovo compito), combinando la prossimità geografica con l'allineamento delle finestre temporali e il tempo di tolleranza (slack) residuo.

1.3 Large Language Models

I large language models (LLM) basano le loro straordinarie capacità sull'architettura Transformer, introdotta per superare i limiti delle reti neurali ricorrenti (RNN) e convoluzionali. I modelli tradizionali, infatti, elaboravano le sequenze di testo in modo strettamente sequenziale (parola per parola), ostacolando la parallelizzazione del calcolo e rendendo difficile l'apprendimento delle dipendenze a lungo raggio. Il Transformer ha rivoluzionato questo paradigma eliminando del tutto la ricorrenza e affidandosi esclusivamente ai meccanismi di attenzione [Vaswani *et al.*, 2017].

- Self-Attention: Meccanismo che pone in relazione tutte le posizioni all'interno di una medesima sequenza per generare una rappresentazione semantica globale. Tale architettura consente di collegare due token qualsiasi attraverso un numero costante di operazioni ($O(1)$), a prescindere dalla loro distanza all'interno della frase.
- Multi-Head Attention: Invece di calcolare un'unica funzione di attenzione, il Transformer proietta linearmente le Query, le Keys e i Values in molteplici sottospazi dimensionali (le "teste" o heads). Questo permette al modello di prestare attenzione simultaneamente a informazioni provenienti da diversi spazi di rappresentazione, catturando parallelamente molteplici relazioni sintattiche e semantiche.

Rappresentazioni vettoriali

Per poter essere processato matematicamente dall'architettura Transformer, il testo libero (composto da parole o "token") deve prima essere mappato in uno spazio vettoriale continuo.

- Text Embeddings: I token di input e di output vengono convertiti in vettori di dimensione fissa tramite matrici di embedding apprese durante l'addestramento. In questo spazio geometrico, concetti linguisticamente simili tendono ad assumere posizioni vicine.
- Positional Encoding: Poiché il Transformer non utilizza alcuna rete ricorrente o convoluzionale, l'architettura di base non ha una nozione intrinseca dell'ordine in cui le parole compaiono nella sequenza. Per rimediare a questo, vengono aggiunti dei codici posizionali direttamente ai vettori di embedding all'ingresso del modello.

1.3.1 Chain-of-Thought

Una tecnica fondamentale per sbloccare le capacità di ragionamento logico dei modelli è il Chain-of-Thought (CoT) Prompting. Invece di richiedere al modello di fornire direttamente la risposta finale, il CoT spinge il modello a generare una serie di passaggi intermedi di ragionamento in linguaggio naturale prima di giungere alla conclusione [Wei *et al.*, 2022]. I vantaggi accademici e operativi di questa tecnica sono molteplici. In primo luogo, il CoT permette ai Large Language Models (LLM) di scomporre problemi a più fasi in passaggi

intermedi, consentendo al modello di allocare una maggiore potenza di calcolo, tradotta in un maggior numero di token generati, per affrontare agevolmente i problemi logici, simbolici o aritmetici più complessi. Inoltre, il successo di questo approccio si manifesta come una vera e propria abilità emergente che dipende criticamente dalla scala del modello. Per i modelli linguistici con pochi parametri, infatti, il CoT non porta alcun beneficio e a volte peggiora persino le prestazioni producendo ragionamenti fluidi ma illogici; al contrario, nei modelli che superano i 100 miliardi di parametri, il prompting CoT garantisce miglioramenti significativi. Infine, un ulteriore e cruciale vantaggio risiede nell'interpretabilità: la catena di pensieri generata funge da vera e propria "finestra interpretabile" sul comportamento del modello. Tale trasparenza permette agli sviluppatori umani di leggere il processo step-by-step, suggerendo come l'intelligenza artificiale sia arrivata a una determinata risposta e offrendo la preziosa opportunità di diagnosticare e correggere il punto esatto in cui il ragionamento ha fallito.

1.3.2 Sistemi Agentici

Il paradigma dell'Agentic AI estende l'impiego degli LLM dalla mera generazione passiva alla risoluzione attiva ed interattiva dei problemi permettendo loro di percepire, ragionare e agire in modo autonomo. Formalmente, un agente basato su LLM opera come un sistema di controllo dinamico all'interno di un processo decisionale di Markov parzialmente osservabile (POMDP) [Arunkumar *et al.*, 2026]. Ad ogni passo temporale t, l'agente riceve un'osservazione parziale O_t dall'ambiente, aggiorna il proprio stato di memoria interno, genera un piano di ragionamento latente Z_t ed esegue infine un'azione esterna A_t la cui policy è esplicitamente condizionata dal ragionamento appena prodotto.

Nonostante il prompting Chain-of-Thought (CoT) abbia sbloccato incredibili capacità deduttive, il modello genera i propri pensieri usando esclusivamente le rappresentazioni interne, senza alcun ancoraggio col mondo esterno. Questa disconnessione dall'ambiente circostante limita la capacità di aggiornare la conoscenza dinamicamente ed espone pesantemente il sistema alle allucinazioni fattuali e alla propagazione degli errori logici a cascata.

Il ReAct framework

Per superare i limiti dell'azione "cieca" o del ragionamento isolato, è stato introdotto il paradigma ReAct, che combina la generazione di tracce di ragionamento verbale e di azioni specifiche per il task in maniera interlacciata e sinergica. ReAct si ispira direttamente alla cognizione umana, unicità basata sulla capacità di fondere azioni orientate all'obiettivo col "dialogo interiore" (ragionamento verbale), essenziale per l'autoregolazione, la strategizzazione e il mantenimento della memoria di lavoro. Dal punto di vista matematico e strutturale, ReAct estende lo spazio degli output generabili dall'agente definendo un contesto combinato $\hat{\mathcal{A}} = \mathcal{A} \cup \mathcal{L}$, dove $\mathcal{A}$ rappresenta l'insieme delle azioni ammissibili nell'ambiente esterno e $\mathcal{L}$ indica lo spazio del linguaggio naturale dedicato ai pensieri di ragionamento. Le traiettorie di esecuzione assumono così la forma $\tau = (o_1, z_1, a_1, o_2, z_2, a_2, \dots)$, in cui le tracce di pensiero $z_t \in \mathcal{L}$ guidano logicamente la selezione delle azioni successive $a_t \in \mathcal{A}$, le quali generano a loro volta le corrispondenti osservazioni ambientali o_{t+1} [Yao *et al.*, 2023].

All'interno di questo framework, si instaura una perfetta sinergia bidirezionale:

- Ragionare per Agire (Reason to act): Le tracce di ragionamento verbale (z_t) permettono al modello di scomporre dinamicamente gli obiettivi di alto livello, creare e tracciare piani d'azione, dedurre conoscenze di buon senso e gestire le eccezioni, correggendo la rotta quando necessario.

- Agire per Ragionare (Act to reason): L'esecuzione di azioni (a_t) permette all'agente di interfacciarsi con l'ambiente esterno (ad es., tramite API, motori di ricerca o sistemi operativi) per raccogliere informazioni mancanti e aggiornare il contesto del problema, supportando così le deduzioni future.

Attraverso il ciclo continuo di percezione-pensiero-azione-osservazione, il processo di risoluzione dei problemi diventa molto più ancorato ai fatti, interpretabile e affidabile. Analisi comparative dimostrano che l'accesso al feedback esterno tramite ReAct abbatte drasticamente il tasso di falsi positivi causato dalle allucinazioni del solo Chain-of-Thought.

Nella Figura 1.3 viene illustrata l'architettura di un sistema agentico.

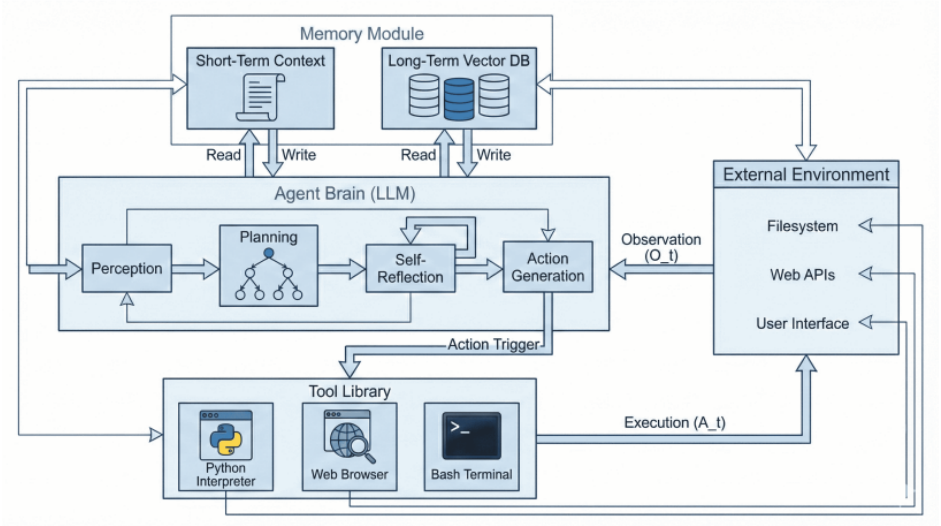


Figura 1.3: Architettura dei sistemi agentici

Controllo, Sicurezza e Grounding

L'integrazione di agenti basati su LLM in contesti operativi critici (come la logistica e la gestione di reti) richiede meccanismi di vincolo per garantire affidabilità, interpretabilità ed esecuzione sicura.

Il concetto di Trust-Region Durante la fase di addestramento o di fine-tuning dell'agente, effettuare modifiche troppo brusche ai parametri del modello può causare un crollo improvviso delle prestazioni. Per evitare questa instabilità, il metodo Trust Region Policy Optimization (TRPO) impone un limite alle variazioni: la nuova strategia (policy) π_θ è costretta a rimanere all'interno di una "regione di confidenza" (trust region) rispetto a quella precedente $\pi_{\theta_{\text{old}}}$. Dal punto di vista matematico, ciò si ottiene ponendo un vincolo sulla divergenza di Kullback-Leibler (KL), che limita la variazione massima consentita entro un valore δ. In questo modo, TRPO garantisce che le prestazioni dell'agente migliorino in modo continuo e graduale, evitando sbalzi di convergenza o peggioramenti imprevisti [Schulman *et al.*, 2015].

Negli scenari operativi di dispatching logistico in tempo reale, questo principio teorico viene tradotto dall'Orchestratore agentico in vincoli geometrici deterministici applicati all'inferenza. Invece di regolare reti neurali ad alta dimensionalità, la Trust-Region viene imposta direttamente sulle variazioni dei parametri operativi per mitigare allucinazioni o comportamenti erratici del Large Language Model.

Grounding Se un agente si affida esclusivamente al proprio ragionamento interno, rischia di generare "allucinazioni" (informazioni inventate ma apparentemente plausibili) e di propagare errori di logica lungo il processo. Il Grounding (ovvero l'ancoraggio alla realtà) indica l'insieme di meccanismi che vincolano i pensieri e le decisioni del modello a dati ed evidenze oggettive provenienti dall'ambiente esterno (come avviene, ad esempio, nel paradigma ReAct). Interagendo con strumenti e dati reali, l'agente può verificare e correggere i propri passaggi logici tramite osservazioni concrete, trasformando il ragionamento in un processo verificabile, sempre aggiornato e perfettamente aderente ai vincoli del dominio operativo.

CAPITOLO 2

Ipergrafo tipizzato e orchestrazione agentica

In questo capitolo si formalizza la risoluzione del Last-Mile Delivery tramite un ipergrafo tipizzato e pesato, integrato con un orchestratore agentico basato su LLM. Dopo aver definito i vincoli del doppio orizzonte decisionale, si descrivono le metriche di diffusione su grafo, le politiche di instradamento locale e le pipeline di sicurezza con vincolo di Trust-Region.

2.1 Architettura generale e orizzonti decisionali

La logistica dell'ultimo miglio è caratterizzata da una tensione strutturale tra due fasi decisionali distinte: la pianificazione statica del mattino e la gestione dinamica delle richieste pomeridiane. Tale dicotomia delinea una naturale configurazione a "doppio orizzonte" (double-horizon).

La prima è la fase mattutina (morning planning), in cui il sistema possiede una conoscenza completa della domanda pregressa. Formalmente, si definisce un insieme di corrieri R, un insieme di zone Z, un insieme di magazzini (depot) H e un set di consegne note a priori $T^{(am)} \subseteq T$. Ogni specifico task $t \in T^{(am)}$ è vincolato matematicamente da una finestra temporale $\tau(t) = [\tau^-(t), \tau^+(t)]$ definita su un orizzonte continuo (ad esempio $[0, 24\,\mathrm{h}]$) e da un requisito fisico di capacità o volume $q(t)$. L'obiettivo primario di questa fase è l'assegnazione dei corrieri a specifiche comunità di lavoro per generare sequenze di rotte ordinate $R_u = \{r_{u,1}, \ldots, r_{u,l_u}\}$ per ciascun veicolo. Questa pianificazione mira a ottimizzare una funzione obiettivo globale che minimizza i costi di viaggio, i tempi di lavoro e le penalità quadratiche per il ritardo delle consegne, nel rigoroso rispetto dei vincoli fisici, assicurandosi che il carico cumulato non superi la capacità massima del veicolo.

A questa stabilità iniziale subentra la fase pomeridiana (afternoon matching), in cui il problema si trasforma in un modello puramente dinamico. Durante l'esecuzione delle rotte, il sistema riceve in tempo reale nuove richieste di ritiro, definendo un secondo insieme di task dinamici $T_{(pm)} \in T$, disgiunto dal precedente tale per cui $T_{(am)} \cap T_{(pm)} = \emptyset$. In questo scenario, l'obiettivo si sposta sulla reattività: per ogni nuovo task t in arrivo, diventa cruciale decidere istantaneamente l'assegnazione e i relativi tempi di partenza senza dover ricalcolare l'intera soluzione. Poiché riapplicare solutori esatti (come i modelli MILP) sull'intero spazio di ricerca combinatorio per ricalcolare globalmente le rotte richiederebbe tempi inaccettabili rispetto ai millisecondi di latenza operativa concessi, l'inserimento dei nuovi task in streaming avviene valutando esclusivamente la fattibilità locale.

2.1.1 Limiti delle rappresentazioni tradizionali a grafo

La modellazione di relazioni complesse tra entità eterogenee (riders, zone, tasks e depositi) tramite le strutture classiche della teoria dei grafi presenta severi limiti computazionali e concettuali:

- Grafi semplici e relazioni binarie: Un grafo classico $G = (V, E)$ ammette esclusivamente archi binari, in grado di connettere coppie di nodi (relazioni 1-a-1). Nella logistica dell'ultimo miglio, tuttavia, le relazioni chiave sono intrinsecamente ad alto ordine (n-arie). Ad esempio, la compatibilità fisica di un rider r_1 che serve una coppia di compiti t_1, t_2 nella zona z_1 richiede di legare simultaneamente quattro entità. Rappresentare questo legame con archi binari richiede l'introduzione di nodi fittizi di raccordo o la frammentazione della relazione in molteplici archi adiacenti, disperdendo l'informazione di coesione dell'insieme e appesantendo la topologia della rete.
- Reti Multilivello: Una rete multilivello segrega le diverse tipologie di entità in strati o livelli distinti (ad esempio, un livello per la flotta, uno per le zone e uno per i compiti). Per consentire la comunicazione tra i livelli, la teoria classica Kivelä *et al.* [2014] impone la duplicazione dei nodi su più livelli (ad esempio, proiettando il nodo del rider sullo strato delle zone) e il tracciamento di archi trasversali (inter-layer). Questa ridondanza genera matrici di adiacenza di dimensioni enormi. Di conseguenza, i calcoli stocastici (come i cammini casuali) richiedono regole ad-hoc per gestire i passaggi di livello, aumentando i tempi di calcolo e i requisiti di memoria.

2.1.2 L'ipergrafo tipizzato

Per superare i limiti strutturali e le inefficienze computazionali dei grafi tradizionali e delle reti multilivello, l'approccio proposto modella il sistema logistico tramite un singolo ipergrafo tipizzato e pesato $\mathcal{H} = (V, \mathcal{E}, w, \theta)$. Dal punto di vista concettuale, questa scelta consente di unire in un unico spazio combinatorio tutte le entità operative del dominio (corrieri, zone geografiche, compiti di consegna e depositi) azzerando la duplicazione dei nodi.

L'elemento qualificante dell'ipergrafo risiede nell'impiego degli iperarchi per codificare le relazioni molti-a-molti (n-arie) che caratterizzano la logistica dell'ultimo miglio. Ad esempio, la compatibilità congiunta tra un rider, una zona di consegna e un insieme di task può essere espressa nativamente da un singolo iperarco, senza ricorrere a nodi artificiali o a pesanti stratificazioni multilayer. Come approfondito nella Sezione 2.2, questa formulazione si traduce in una matrice di incidenza sparsa ad aggiornamento rapido, ideale per le strette finestre temporali dello streaming pomeridiano.

Inoltre, la suddivisione degli iperarchi in categorie tipizzate definisce una famiglia di cammini casuali di tipo Vertex-Hyperedge-Vertex (V-E-V) regolati da pesi e temperature specifiche. Questa modellazione offre il punto di contatto ideale per l'Orchestratore LLM, agendo esclusivamente sui parametri di temperatura delle diverse tipologie di iperarchi, l'agente può calibrare dinamicamente le priorità del sistema e adattare l'instradamento in tempo reale, senza mai dover modificare la topologia fisica sottostante o invocare solutori combinatori pesanti.

2.2 Costruzione dell'ipergrafo tipizzato

Per tradurre lo scenario operativo del last-mile delivery in una rappresentazione matematica unificata, si definisce l'ipergrafo tipizzato e pesato $\mathcal{H} = (V, \mathcal{E}, w, \theta)$. La costruzione

dell'ipergrafo avviene a partire da dati reali del sistema di trasporto e si articola nella definizione dei vertici, nella tassonomia degli iperarchi e nella normalizzazione algebrica dei pesi associati.

2.2.1 Vertici, iperarchi e matrice di incidenza

L'insieme dei vertici V è partizionato in quattro classi eterogenee che rappresentano le entità fisiche ed operative coinvolte nel processo logistico:

$$V = R \cup Z \cup T \cup H$$

dove R indica l'insieme dei corrieri (riders), Z l'insieme delle zone geografiche stradali, T l'insieme dei task di consegna (deliveries) e pickup, e H l'insieme dei depositi fisici. Le relazioni n-arie tra queste entità sono codificate da un insieme di iperarchi $\mathcal{E}$, dove ciascun iperarco $e \in \mathcal{E}$ è un sottoinsieme non vuoto di nodi ($e \subseteq V$) [Bretto, 2013]. L'accoppiamento strutturale tra nodi e iperarchi viene espresso matematicamente tramite la matrice di incidenza generalizzata $X \in \{0,1\}^{|V|\times|\mathcal{E}|}$. Ciascun elemento $X_{v,e}$ della matrice di incidenza è definito in base all'appartenenza del nodo all'iperarco:

$$X_{v,e} = \begin{cases} 1 & \text{se } v \in e \\ 0 & \text{altrimenti} \end{cases}$$

Questa matrice binaria e sparsa costituisce l'unica rappresentazione combinatoria di base del sistema logistico . Per integrare l'eterogeneità delle relazioni, la struttura topologica viene affiancata da pesi e temperature. Nello specifico, a ciascun iperarco e è associato un peso normalizzato w_e e un parametro di temperatura $\gamma_{\theta(e)} \in (0,1]$ specifico del tipo di iperarco. Durante i processi di esplorazione e propagazione stocastica, l'influenza relativa dei diversi strati relazionali viene calibrata dinamicamente definendo un peso di transizione verso l'iperarco pari a: $\omega(e) = w_e\gamma_{\theta(e)}$. Questo approccio permette di modulare l'importanza operativa delle diverse relazioni senza mai alterare la matrice di incidenza fondamentale X.

2.2.2 Tassonomia matematica dei cinque iperarchi

Per riflettere i diversi vincoli fisici, logistici e temporali del dominio reale, si definiscono cinque tipologie di iperarchi tipizzati tramite una mappa di classificazione $\theta : \mathcal{E} \to \Theta$, con $\Theta = \{\text{ctx}, \text{fam}, \text{fea}, \text{dep}, \text{tmp}\}$.

Iperarchi di Contesto (e^{ctx})

Questi iperarchi codificano la domanda locale associando una specifica zona geografica all'insieme di consegne previste al suo interno durante una data finestra temporale. Dividendo l'orizzonte di pianificazione in intervalli temporali disgiunti B_b, l'insieme dei task per una zona z in quel lasso di tempo è definito come $T_{z,b}$. L'iperarco di contesto viene formalizzato come l'unione tra il nodo della zona e i relativi task:

$$e^{\text{ctx}}_{z,b} = \{z\} \cup T_{z,b}$$

Il suo peso iniziale riflette la densità della domanda locale, risultando proporzionale al rapporto tra il numero di task e l'area geografica della zona:

$$w(e^{\text{ctx}}_{z,b}) = f_{\text{ctx}}\left(\frac{|T_{z,b}|}{\text{area}(z)}\right)$$

dove la funzione monotona f_{ctx} applica una trasformazione logaritmica seguita da un riscalamento lineare min-max per ricondurre i valori nell'intervallo $[0,1]$.

Iperarchi di Familiarità (e^{fam})

Rappresentano la competenza territoriale maturata storicamente da ciascun rider. Per ogni coppia rider $r \in R$ e zona $z \in Z$ per cui esista uno storico di consegne completate con successo, si definisce l'iperarco binario:

$$e^{\text{fam}}_{r,z} = \{r, z\}$$

Il peso viene calcolato applicando un decadimento temporale esponenziale agli eventi storici passati, privilegiando i successi recenti:

$$w(e^{\text{fam}}_{r,z}) = f_{\text{fam}}\left(\sum_{i=1}^{N_{r,z}} \kappa^{\Delta_i} y_i\right) \quad \text{con } \kappa \in (0,1)$$

dove $y_i \in \{0,1\}$ indica il successo dell'operazione, Δ_i rappresenta l'età dell'evento espressa in ore, e κ è il tasso di decadimento temporale. In presenza di feedback qualitativi dal campo, a questo iperarco può essere associato anche un segno $s(e^{\text{fam}}_{r,z}) \in \{-1, +1\}$ per distinguere i giudizi positivi da quelli negativi.

Iperarchi di Fattibilità (e^{fea})

Vincolano l'operatività fisica del corriere. Definito l'insieme $S_{r,z}$ dei task associati alla zona z che il rider r può legalmente e fisicamente servire in base alle finestre orarie del suo shift $\sigma(r)$, alla capacità massima $\text{cap}(r)$ ed a eventuali permessi di accesso:

$$S_{r,z} = \{t \in T : \pi(t) = z,\, q(t) \leq \text{cap}(r),\, \tau(t) \cap \sigma(r) \neq \emptyset,\, \text{legal}(r,t) = 1\}$$

Se $S_{r,z} \neq \emptyset$, viene istanziato l'iperarco molti-a-molti:

$$e^{\text{fea}}_{r,z} = \{r, z\} \cup S_{r,z}$$

Al fine di garantire la necessaria flessibilità per i successivi inserimenti dinamici nel pomeriggio, il peso dell'iperarco $w(e^{\text{fea}}_{r,z})$ applica una penalizzazione soft proporzionale all'occupazione del veicolo, evitando così la saturazione prematura delle risorse in fase mattutina. Matematicamente, il peso combina il numero di task compatibili con un fattore lineare di smorzamento del carico g_{load}:

$$w(e^{\text{fea}}_{r,z}) = f_{\text{fea}}(|S_{r,z}|) \cdot g_{\text{load}}\left(\frac{\sum_{t \in S_{r,z}} q(t)}{\text{cap}(r)}\right) \tag{2.2.1}$$

dove, definito il tasso di occupazione del mezzo $u = \frac{\sum_{t \in S_{r,z}} q(t)}{\text{cap}(r)} \in [0,1]$, la funzione di penalizzazione è formalizzata come:

$$g_{\text{load}}(u) = \max\{\epsilon_0,\, 1 - u\}, \quad \text{con } \epsilon_0 > 0 \tag{2.2.2}$$

In questa formulazione, la funzione monotona f_{fea} modula la forza del legame al crescere dei compiti servibili $|S_{r,z}|$, mentre il termine $g_{\text{load}}(u) \in (0,1]$ applica uno smorzamento progressivo all'avvicinarsi della saturazione volumetrica, garantendo al contempo che il peso non decada mai a zero (grazie al valore di guardia ϵ_0) per preservare la connettività strutturale dell'iperarco.

Iperarchi di Copertura Deposito (e^{dep})

Mappano la raggiungibilità delle zone di consegna a partire dai depositi stradali fisici. Per ogni deposito $h \in H$ e zona $z \in Z$, si definisce l'iperarco:

$$e^{\text{dep}}_{h,z} = \{h, z\}$$

Il cui peso è inversamente proporzionale alla distanza stradale minima $dist_Z(h, z)$ misurata sul grafo delle zone:

$$w(e^{\text{dep}}_{h,z}) = f_{\text{dep}}\left(\frac{1}{\epsilon + dist_Z(h, z)}\right) \quad \text{con } \epsilon > 0$$

dove $dist_Z$ rappresenta la distanza calcolata tra i confini delle zone sul grafo stradale e ϵ è un piccolo termine introdotto per garantire la stabilità numerica.

Grafo di adiacenza delle zone Per mantenere l'efficienza strutturale, la prossimità geometrica tra le zone operative non viene modellata come uno strato separato (layer) dell'ipergrafo, ma agisce come conoscenza geografica a priori per guidare le transizioni stocastiche. A tale scopo, si definisce un grafo di adiacenza delle zone $G_Z = (Z, E_Z)$ proiettando la rete stradale reale sulla zone della città. La distanza spaziale tra due zone $z, z' \in Z$ è definita matematicamente come la minima distanza stradale tra i loro rispettivi confini:

$$\text{dist}_Z(z, z') = \min_{u \in \partial z,\, v \in \partial z'} \text{dist}_{\text{road}}(u, v) \tag{2.2.3}$$

dove ∂z denota l'insieme dei punti stradali sul confine della zona z, e $\text{dist}_{\text{road}}(u, v)$ rappresenta la metrica del cammino minimo calcolata sulla rete stradale. Gli archi (z, z') vengono aggiunti al grafo G_Z se e solo se la distanza $\text{dist}_Z(z, z')$ è finita e inferiore ad una specifica soglia di calibrazione. Le distanze spaziali così ottenute determinano i costi di transizione per le camminate casuali e per il calcolo delle valutazioni di inserimento locale basate sulla prossimità geografica.

Iperarchi Temporali (e^{tmp})

Raggruppano i compiti di consegna in base all'urgenza e alla pressione delle scadenze all'interno di uno specifico intervallo (B_b) in cui devono essere serviti. Dopo aver definito la "pressione" (tightness) $\text{tight}(t) = \max\{0, \tau^+(t) - \tau^-(t) - \hat{s}(t)\}$ di una consegna come il tempo di margine utile calcolato sottraendo dalla finestra temporale il buffer di viaggio e servizio stimato $\hat{s}(t)$, l'iperarco raccoglie tutti i task la cui finestra operativa interseca l'intervallo:

$$e^{tmp}_b = \{t \in T : \tau(t) \cap B_b \neq \emptyset\} \tag{2.2.4}$$

Il cui peso è regolato inversamente dalla rigidità temporale media (tightness) dei compiti contenuti, esaltando in modo naturale l'urgenza dei task critici:

$$w(e^{tmp}_b) = f_{tmp}\left(\frac{1}{\epsilon + \text{avg}_{t \in e^{tmp}_b} \text{tight}(t)}\right) \tag{2.2.5}$$

In questo modo, i time-bin contenenti ordini urgenti e con finestre orarie stringenti esercitano un'influenza stocastica significativamente maggiore durante le diffusioni su grafo.

2.2.3 Commensurabilità dei pesi e gestione della sparsità

Al fine di rendere commensurabili i pesi generati da segnali eterogenei (come conteggi, distanze e affinità), il modello applica una normalizzazione a due fasi condotta separatamente all'interno di ciascun tipo di iperarco. I segnali grezzi subiscono prima un log-scaling per attenuare l'effetto delle distribuzioni a coda lunga (evitando che poche zone iper-frequentate monopolizzino i calcoli), e vengono successivamente riscalati linearmente nell'intervallo $[0, 1]$. Inoltre, per limitare l'occupazione di memoria e preservare la sparsità della matrice di incidenza, il sistema applica meccanismi di potatura: gli iperarchi con un peso normalizzato inferiore a una specifica soglia di tipo vengono omessi, e la cardinalità massima degli iperarchi di fattibilità viene limitata trattenendo unicamente i task più urgenti (valutati in base alla loro "pressione" temporale).

2.3 Grafi derivati e transizioni markoviane

Per interrogare ed esplorare la struttura n-aria dell'ipergrafo tipizzato $\mathcal{H}$, si definiscono tre costruzioni grafiche ausiliarie ed un processo stocastico markoviano a due stadi calcolato direttamente sulla topologia dell'ipergrafo.

2.3.1 Grafi ausiliari e proiezioni tipizzate

L'esplorazione delle relazioni dell'ipergrafo si avvale di tre costruzioni grafiche ausiliarie, ciascuna progettata per supportare specifiche analisi.

In primo luogo, si definisce il grafo di incidenza $G_I = (V \cup \mathcal{E}, E_I)$, una struttura bipartita in cui un arco $(v, e) \in E_I$ esiste se e solo se l'elemento corrispondente della matrice soddisfa $X_{v,e} = 1$. In questo grafo, i pesi degli archi ereditano direttamente i pesi degli iperarchi w_e e non sono soggetti a una parametrizzazione indipendente.

In secondo luogo, per analizzare la struttura mesoscopica del sistema, si introduce il Line Graph degli iperarchi $L(\mathcal{E})$, costruito direttamente sull'insieme degli iperarchi $\mathcal{E}$. In $L(\mathcal{E})$, due iperarchi e ed e' sono collegati da un arco non orientato ogni volta che condividono almeno un vertice comune ($e \cap e' \neq \emptyset$). Il peso di ciascun arco viene formalizzato come:

$$W_{ee'} = w_e \cdot w_{e'} \frac{|e \cap e'|}{|e \cup e'|} \tag{2.3.1}$$

Questa formulazione permette di bilanciare la forza del legame relazionale con l'ampiezza della sovrapposizione tra iperarchi, producendo una struttura sparsa e ben condizionata per gli algoritmi di community detection ed evitando la formazione di clique spurie indotte da iperarchi ad elevata cardinalità.

Infine, le proiezioni tipizzate consentono di restringere l'incidenza a selezionate classi di vertici, evitando di collassare l'informazione in modo asimmetrico. Nello specifico, il grafo bipartito Rider–Zona (G_{RZ}) presenta una matrice di adiacenza pesata descritta da:

$$A_{RZ}(r, z) = \sum_{e \in \mathcal{E} : \{r,z\} \subset e} \frac{\lambda_{\theta(e)} \cdot w_e}{|e|} \tag{2.3.2}$$

dove il fattore $\lambda_{\theta(e)} \in [0, 1]$ agisce da peso di tipo per smorzare l'influenza degli iperarchi la cui elevata cardinalità dominerebbe altrimenti la proiezione. Nella pratica operativa, la maggior parte della massa viene assegnata ai parametri λ_{fam} e λ_{fea}, in quanto codificano in modo diretto la competenza storica del corriere e la fattibilità legale delle consegne.

Parallelamente, il grafo di co-occorrenza dei Task (G_{TT}) adotta la formulazione:

$$A_{TT}(t,t') = \sum_{e\in\mathcal{E}:\{t,t'\}\subset e} \frac{\lambda_{\theta(e)}\cdot w_e}{|e|} \tag{2.3.3}$$

la quale viene impiegata esclusivamente per la risoluzione di query di similarità tra compiti e mai per l'imposizione di vincoli rigidi di instradamento.

2.3.2 Kernel di cammino casuale Vertex–Hyperedge–Vertex (V-E-V)

I cammini casuali sono definiti sull'ipergrafo originale anziché su una proiezione per preservare la natura relazionale molti-a-molti. Si impiega un kernel privo di memoria a due stadi che alterna vertici e iperarchi.

Nel primo stadio, relativo alla transizione da vertice ad iperarco ($V \to \mathcal{E}$), per un vertice v incidente all'insieme di iperarchi $\mathcal{E}(v) = \{e \in \mathcal{E} : X_{v,e} = 1\}$, la probabilità di transizione verso l'iperarco $e \in \mathcal{E}(v)$ è definita come:

$$P(v \to e) = \frac{\omega(e)}{\sum_{e'\in\mathcal{E}(v)}\omega(e')}, \quad \text{con } \omega(e) = w_e \cdot \gamma_{\theta(e)} \tag{2.3.4}$$

dove $\gamma_{\theta(e)} \in (0,1]$ rappresenta la temperatura di tipo che riduce l'influenza di categorie ad alta cardinalità o con bassa fedeltà.

Nel secondo stadio, concernente la transizione da iperarco a vertice ($\mathcal{E} \to V'$), la probabilità che dall'iperarco e si ritorni al vertice $v' \in e$ è espressa da:

$$P(e \to v') = \frac{\phi(v',e)}{\sum_{u\in e}\phi(u,e)}, \quad \text{con } \phi(v',e) = \psi_{\theta(e)}(v',e)\cdot\eta(v',e) \tag{2.3.5}$$

dove la funzione $\psi_{\theta(e)}(v',e)$ definisce le preferenze intrinseche legate alla specifica tipologia dell'iperarco $\theta(e)$, mentre il fattore combinato $\eta(v',e) = K_Z(\cdot)\cdot\rho(\cdot)$ accoppia in modo sinergico la dimensione spaziale e quella temporale. Nello specifico, la componente di località geografica sfrutta la conoscenza a priori del grafo di adiacenza delle zone G_Z: quando il vertice di destinazione v' e almeno un altro nodo appartenente all'iperarco e sono ancorati a specifiche zone geografiche (rispettivamente z' e z), viene applicato un kernel di attenuazione esponenziale basato sulla distanza stradale, mentre in assenza di riferimenti alle zone all'interno dell'iperarco il fattore assume il valore neutro predefinito $K_Z = 1$. Parallelamente, la compatibilità temporale modula la probabilità di transizione in base all'urgenza operativa, applicando un fattore di scala $\rho(t,e) \in (0,1]$ qualora il nodo v' sia un task $t \in e$ caratterizzato da una finestra temporale ristretta o prossima alla violazione; per tutte le entità differenti dai task (come corrieri, depositi o zone), il vincolo temporale non trova applicazione e il fattore assume valore unitario $\rho = 1$

Combinando i due stadi, il kernel markoviano vertice-a-vertice risultante è:

$$P(v \to v') = \sum_{e\in\mathcal{E}} \mathbb{I}[v\in e]\cdot\mathbb{I}[v'\in e]\cdot P(v\to e)\cdot P(e\to v') \tag{2.3.6}$$

Per garantire l'ergodicità e l'aperiodicità della catena e consentire il calcolo di punteggi personalizzati, viene introdotto un meccanismo di riavvio con vettore di personalizzazione ρ e parametro $\alpha \in (0,1)$:

$$\tilde{P}(v \to v') = \alpha\cdot\rho(v') + (1-\alpha)\cdot P(v\to v') \tag{2.3.7}$$

L'associazione con il Personalized PageRank (PPR) su G_{RZ} si ottiene restringendo ρ a corrieri e zone ed azzerando $\gamma_{\theta(e)}$ e $\psi_{\theta(e)}$ per i tipi irrilevanti alla struttura bipartita.

2.3.3 Formulazione matriciale

Ai fini dell'implementazione computazionale e dell'analisi di complessità, il kernel viene espresso in forma matriciale compatta. Definita la matrice di incidenza pesata $H = X \cdot \mathrm{diag}(w)$, si introducono la matrice diagonale delle temperature $G_E = \mathrm{diag}(\gamma_{\theta(e)})$ e la matrice diagonale dei gradi scalati $D_V = \mathrm{diag}(HG_E\mathbf{1})$, oltre alla mappa stocastica per riga iperarco-vertice $\Pi_{E\to V}$ contenente gli elementi $P(e \to v')$.

Il kernel vertice-vertice stocastico per righe privo di riavvio assume la seguente forma matriciale sparsa:

$$P = D_V^{-1} H G_E \Pi_{E\to V} \tag{2.3.8}$$

Tutti i fattori costituenti sono sparsi e la matrice $\Pi_{E\to V}$ può essere aggiornata localmente in caso di modifiche limitate agli iperarchi, fornendo la base algoritmica per gli aggiornamenti incrementali in fase di streaming.

2.4 Le metriche principali

Per evitare l'esplorazione esaustiva dell'albero combinatorio delle soluzioni, il sistema sintetizza la dinamica logistica attraverso cinque funzioni analitiche. Queste metriche estraggono segnali dal kernel markoviano della rete e dalle sue proiezioni, fungendo da surrogati analitici per guidare le decisioni sia nella pianificazione statica del mattino sia nello streaming dinamico del pomeriggio.

2.4.1 Misure di affinità e compatibilità

Le prime due metriche modellano quantitativamente la competenza territoriale dei corrieri e la compatibilità diretta tra corrieri e singoli compiti, garantendo un'assegnazione altamente specializzata.

Driver-Area Affinity (DAA)

La Driver-Area Affinity quantifica la competenza e l'affinità territoriale di un corriere $r \in R$ nei confronti di una determinata zona $z \in Z$. L'intuizione alla base di questa misura risiede nella diffusione probabilistica: immaginando di rilasciare una particella stocastica sul nodo del corriere, la densità locale degli iperarchi di familiarità e fattibilità tenderà a trattenerla nelle aree in cui il corriere possiede maggiore esperienza o compatibilità. Matematicamente, la metrica si ottiene calcolando la distribuzione stazionaria $\mathfrak{ß}_r$ di un cammino casuale con riavvio personalizzato (Personalized PageRank):

$$\mathfrak{ß}_r^\top = \alpha \mathfrak{æ}_r^\top + (1-\alpha)\mathfrak{ß}_r^\top P \tag{2.4.1}$$

dove P è il kernel markoviano della rete, $\alpha \in (0,1)$ rappresenta la probabilità di riavvio e $\mathfrak{æ}_r$ è il vettore di personalizzazione che assegna massa unitaria al nodo del corriere r. Poiché corrieri diversi presentano storici di diversa intensità, la componente della distribuzione relativa alle sole zone, $[\mathfrak{ß}_r]_z$, viene sottoposta a una normalizzazione di scala ristretta alla partizione delle zone (Z-slice):

$$\mathrm{DAA}_r(z) = \frac{[\mathfrak{ß}_r]_z}{\sum_{z'\in Z}[\mathfrak{ß}_r]_{z'}} \tag{2.4.2}$$

Questo passaggio rende i punteggi direttamente confrontabili tra tutti i corrieri, fornendo al sistema una mappa di specializzazione spaziale sparsa e robusta.

Meta-Hyperpath Affinity (MHA)

Se la DAA individua l'area ideale per un corriere, la Meta-Hyperpath Affinity valuta la compatibilità puntuale tra un corriere r e uno specifico compito di consegna o ritiro t. Anziché collassare le relazioni dell'ipergrafo in un semplice grafo binario, la MHA valuta la probabilità di percorrere la catena concettuale di tipo $R \Rightarrow Z \Rightarrow T$. Il calcolo sfrutta il prodotto tra la matrice di transizione corriere-zona M_{RZ} e quella zona-task M_{ZT}. Per evitare che le zone ad altissima densità o centralità (gli "hub" della città) attirino sproporzionatamente tutti i punteggi, si applica una normalizzazione simmetrica nello spazio intermedio, concettualmente analoga alla similarità del coseno:

$$\mathrm{MHA}(r,t) = \left[D_R^{-1/2} M_{RZ} D_Z^{-1} M_{ZT} D_T^{-1/2} \right]_{rt} \tag{2.4.3}$$

in cui D_R, D_Z, D_T indicano le matrici diagonali delle somme per riga dei rispettivi blocchi. In questo modo si ottiene una misura bilanciata che riflette la compatibilità effettiva tra la storia del corriere e le caratteristiche del singolo compito.

2.4.2 Partizionamento spaziotemporale e PDCC-H

La gestione della coesione geografica al mattino e l'inserimento reattivo di nuove consegne nel pomeriggio richiedono metriche in grado di valutare la struttura mesoscopica e il margine temporale residuo.

Community Structure su $L(\mathcal{E})$

Per la pianificazione delle consegne del mattino, il sistema necessita di raggruppare compiti e zone in "pacchetti di lavoro" (working sets) geograficamente e temporalmente coerenti, evitando sovrapposizioni inefficienti tra i corrieri. Tale obiettivo viene raggiunto individuando la struttura in comunità dell'ipergrafo. L'analisi non viene condotta sui singoli nodi, bensì sul Line Graph degli iperarchi $L(\mathcal{E})$, applicando l'algoritmo di Leiden basato sul Constant Potts Model (CPM) [Traag *et al.*, 2019]. La forza delle connessioni nel Line Graph è regolata dal peso d'intersezione:

$$W_{ee'} = w_e \cdot w_{e'} \frac{|e \cap e'|}{|e \cup e'|} \tag{2.4.4}$$

Le comunità di iperarchi così identificate vengono infine proiettate sui nodi sottostanti, definendo sottoinsiemi di zone $Z(C)$ e di task $T(C)$ ben delimitati. Questo partizionamento riduce lo spazio di ricerca e garantisce che ogni corriere operi all'interno di un modulo territoriale compatto.

Pickup-Delivery Coupling Centrality (PDCC-H)

Durante il pomeriggio, quando la flotta è già in viaggio, l'arrivo in streaming di nuove richieste di ritiro impone decisioni di inserimento rapide. La Pickup-Delivery Coupling Centrality stima la capacità di un corriere in servizio r di assorbire una nuova fermata t_p nella zona z_{t_p} senza causare ritardi a catena sulle consegne già programmate. La metrica valuta anzitutto lo slack temporale residuo del corriere nelle zone z del suo percorso:

$$\mathrm{slack}(z) = \max\left\{0, \min_{t \in T(z)} \left[\tau^+(t) - \hat{\mathrm{ETA}}_r(z) - \hat{s}(t)\right]\right\} \tag{2.4.5}$$

dove $\hat{\mathrm{ETA}}_r(z)$ indica l'orario di arrivo stimato nella zona e $\hat{s}(t)$ è il buffer per il servizio ed il tragitto. La centralità di accoppiamento viene quindi calcolata rapportando il margine

temporale alla distanza stradale sul grafo delle adiacenze G_Z e penalizzando gli eventuali disallineamenti orari:

$$\text{PDCC-H}(t_p, r) = \max_{z \in N_Z(r)} \frac{\text{slack}(z)}{\varepsilon + \text{dist}_Z(z, z_{t_p})} \cdot \phi\left(\text{overlap_time}(t_p, r)\right) \tag{2.4.6}$$

con $\varepsilon > 0$ e dove $\phi \in [0, 1]$ misura la sovrapposizione tra la finestra del ritiro e il passaggio del corriere. Un punteggio elevato segnala la presenza di un corriere vicino, con sufficiente margine di tempo ed eccellente compatibilità oraria.

2.4.3 Text-Informed Centrality (TIMC)

Informazioni operative cruciali come "accesso banchina chiuso dopo le 16:00", "codice citofono richiesto" o "limitazioni ZTL" appaiono spesso sotto forma di note a testo libero $\xi(v)$, risultando invisibili alla pura geometria di rete. La Text-Informed Centrality integra questi segnali non strutturati all'interno dell'instradamento. Le note vengono convertite in vettori di embedding $u(v)$ tramite un modello linguistico leggero ed aggregate con il vettore del contesto operativo del turno q per ricavare la salienza semantica:

$$s(v) = \frac{u(v)^\top q}{\|u(v)\|\|q\|} \tag{2.4.7}$$

La centralità finale del nodo emerge da una combinazione convessa tra la centralità topologica pura $C^{\text{hyper}}(v)$ (ottenuta dai cammini casuali sull'ipergrafo) e la salienza semantica $s(v)$:

$$\text{TIMC}(v) = \mu C^{\text{hyper}}(v) + (1 - \mu)s(v), \quad \mu \in [0, 1] \tag{2.4.8}$$

Il parametro μ permette di modulare l'impatto delle informazioni testuali sulla priorità di instradamento. Qualora la nota testuale sia assente, $s(v)$ assume un valore neutro di classe e la metrica regredisce in modo fluido alla pura centralità strutturale della rete.

2.5 Politiche Decisionali

Le metriche ricavate dall'ipergrafo costituiscono l'infrastruttura analitica con cui il sistema prende decisioni operative concrete. Il framework traduce la conoscenza topologica in politiche decisionali di instradamento e assegnazione guidate dal grafo. Queste politiche, pur operando in modo miope per garantire bassa latenza, garantiscono l'ammissibilità fisica dei percorsi verificando sistematicamente i vincoli di capacità e di finestra temporale sul grafo di adiacenza delle zone G_Z.

2.5.1 Assegnazione delle comunità e bilanciamento dei carichi

La pianificazione del mattino inizia convertendo le comunità di iperarchi identificate sul Line Graph $L(\mathcal{E})$ in insiemi di lavoro operativi, definiti dalle zone $Z(C)$ e dai compiti $T(C)$ coinvolti. Il carico di lavoro atteso per una comunità viene stimato sommando i tempi previsti di tragitto e servizio: $\hat{W}(C) = \sum_{t \in T(C)} \hat{s}(t)$. Ciascun corriere r dispone a sua volta di un budget temporale $bud(r)$, derivato dal proprio turno di lavoro, e di una capacità volumetrica residua $cap_{res}(r)$.

L'accoppiamento tra corrieri e comunità avviene mediante un algoritmo di copertura greedy guidato dallo punteggio $Score(r, C)$, il quale bilancia l'affinità territoriale media con una penalità per lo sbilanciamento delle risorse:

$$Score(r, C) = \frac{1}{|Z(C)|} \sum_{z \in Z(C)} \text{DAA}_r(z) - \eta \cdot imbalance(r, C) \tag{2.5.1}$$

dove $\eta \geq 0$ è un parametro di regolazione. La funzione di sbilanciamento $imbalance(r, C)$ valuta lo sforzo richiesto al corriere rispetto alle sue disponibilità residue di tempo e capacità veicolare:

$$imbalance(r, C) = \max\left\{0, \frac{\hat{W}(C)}{\varepsilon + bud(r)} - 1\right\} + \lambda_{cap} \max\left\{0, \frac{\sum_{t \in T(C)} q(t)}{\varepsilon + cap_{res}(r)} - 1\right\} \quad (2.5.2)$$

con λ_{cap} e $\varepsilon > 0$ volti a bilanciare le componenti e prevenire divisioni per zero. Le coppie (r, C) ammissibili con punteggio massimo vengono selezionate iterativamente. Qualora una comunità risulti troppo estesa per essere assegnata ad un singolo corriere senza violare i vincoli di carico, il sistema esegue una bipartizione condotta lungo il taglio di conduttanza minima (tramite la scomposizione di Kernighan-Lin [Kernighan e Lin, 1970]), dividendo la comunità in due sotto-moduli più piccoli e riprendendo l'assegnazione greedy senza frammentare inutilmente le zone.

2.5.2 Pianificazione della rotta locale

Una volta assegnata una comunità ad un corriere r, la sequenza esatta di fermate viene generata senza ricorrere al calcolo esatto, bensì costruendo un cammino guidato dalla comunità. Inizializzato il nodo di ancoraggio corrente z_{curr} nella zona più vicina al deposito del corriere sul grafo G_Z, il compito successivo $t^* \in T(C)$ viene scelto in modo miope massimizzando una funzione obiettivo multi-criterio:

$$t^* = \arg\max_{t \in T(C) \setminus S} \left[w_1 \frac{1}{\varepsilon + \mathrm{dist}_Z(z_{curr}, z_t)} + w_2 \mathrm{closeness}_C(z_t) + w_3 \mathrm{DAA}_r(z_t) - \pi(t; z_{curr}, r) \right] \quad (2.5.3)$$

dove S rappresenta l'insieme delle consegne già pianificate e $\mathrm{closeness}_C(z_t)$ indica la centralità di vicinanza calcolata sul sottografo locale indotto da $Z(C)$. Il termine $\pi(t) = \lambda_{tw} \max\{0, \hat{\mathrm{ETA}}_r(z_t) - \tau^+(t)\}$ applica una penalità progressiva in caso di ritardo rispetto alla scadenza oraria $\tau^+(t)$, stima l'orario di arrivo $\hat{\mathrm{ETA}}$ lungo il percorso.

I pesi decisionali $w_1, w_2, w_3 \in [0, 1]$ (con $\sum w_i = 1$) regolano l'equilibrio tra prossimità spaziale, centralità e affinità storica. Per impedire oscillazioni o scomposizioni repentine nella pianificazione delle fermate durante l'aggiornamento dei parametri, le variazioni dei pesi vengono vincolate all'interno di una regione di fiducia in norma L_1:

$$\|w - w_{prev}\|_1 \leq \delta \quad (2.5.4)$$

Inoltre, una finestra temporale di osservazione pregressa blocca il riordinamento delle fermate già consolidate, garantendo stabilità operativa alle rotte generate.

2.5.3 Inserimento dinamico in streaming

Durante la fase pomeridiana, i nuovi ritiri t_p rivelati in tempo reale vengono gestiti come eventi in streaming. Per ogni nuova richiesta localizzata nella zona z_{t_p}, il sistema valuta l'insieme dei corrieri attivi A calcolando un indice di priorità composito:

$$prio(t_p, r) = \beta_1 \mathrm{DAA}_r(z_{t_p}) + \beta_2 \mathrm{MHA}(r, t_p) + \beta_3 \frac{1}{\varepsilon + \mathrm{dist}_Z(z_{curr}(r), z_{t_p})} - \beta_4 load(r) \quad (2.5.5)$$

dove $z_{curr}(r)$ rappresenta l'ancora corrente del corriere e $load(r) = \sum q(\cdot)/cap(r)$ indica il suo livello di saturazione volumetrica.

Il compito viene preso in considerazione per un numero ristretto k di corrieri candidati aventi la priorità più alta. Per ciascun candidato, il sistema tenta l'inserimento locale del

ritiro nella sequenza di fermate correnti, scegliendo la posizione che minimizza il costo chilometrico incrementale su G_Z, a condizione che la deviazione non violi le finestre temporali delle consegne successive. Se nessun corriere attivo può assorbire la fermata, viene attivato un corriere di riserva r^+ selezionato in base alla massima affinità $\mathrm{DAA}_{r^+}(z_{t_p})$.

Per eliminare il fenomeno del thrashing, ovvero la continua variazione dell'ordine delle fermate che disorienterebbe il corriere mentre è alla guida, si applica il meccanismo della Freeze Window: tutti i segmenti di rotta la cui esecuzione è prevista entro i successivi Δ minuti vengono congelati e resi immodificabili dagli eventi in streaming. Dopo ogni inserimento confermato, gli iperarchi di familiarità e i vettori di diffusione PPR associati vengono aggiornati localmente in tempo reale, ricalcolando la struttura delle comunità unicamente sulle componenti connesse modificate.

2.6 Controllo agentico e guardrail di sicurezza

Mentre l'ipergrafo gestisce la struttura combinatoria e l'efficienza geometrica delle rotte, la natura dinamica della logistica dell'ultimo miglio richiede di integrare informazioni qualitative non strutturate. Per evitare di appesantire la rete con regole euristiche rigide, il sistema introduce un orchestratore basato su un modello linguistico (LLM) leggero. L'agente agisce come un supervisore ad alto livello: non calcola mai percorsi o assegnazioni dirette dei veicoli, ma modula i parametri decisionali del planner tramite piccoli adattamenti controllati (nudges) e punteggi di salienza semantica. Per garantire l'affidabilità ed eliminare i rischi legati ad allucinazioni o instabilità del modello linguistico, l'interazione è regolata da un'architettura di sicurezza deterministica.

2.6.1 Ruolo dell'Orchestratore LLM

L'Orchestratore LLM funge da interfaccia tra il contesto operativo e la topologia dell'ipergrafo. L'agente riceve in ingresso tre flussi informativi: le note a testo libero $\xi(\cdot)$ associate ai nodi, i feedback qualitativi registrati a livello di corriere-zona o corriere-task, e una stringa di direttiva strategica che definisce le priorità del turno (ad esempio, incrementare la tolleranza sulle finestre temporali o privilegiare la saturazione dei veicoli).

Anziché tentare di risolvere il problema combinatorio di instradamento, operazione in cui i modelli linguistici risultano inaffidabili e soggetti ad allucinazioni spaziali, l'LLM limita il proprio raggio d'azione a tre sole uscite quantitative:

1. La generazione dei punteggi di salienza semantica $s(v)$, derivati dagli embeddings testuali usati nella metrica TIMC.

2. Modesti aggiustamenti ai pesi statici w_e di specifici iperarchi per riflettere variazioni temporanee di rischio stradale o restrizioni di accesso.

3. Piccole variazioni parametriche $\Delta\theta$ applicate al vettore dei parametri di scoring e combinazione del sistema:

$$\theta = \left(\alpha, \beta_1, \beta_2, \beta_3, \beta_4, w_1, w_2, w_3, \eta, \lambda_{cap}, \lambda_{tw}\right) \tag{2.6.1}$$

In questo modo, la logistica rimane guidata dalla fisica e dalla geometria dell'ipergrafo, mentre l'LLM apporta la flessibilità contestuale necessaria per gestire eccezioni descrittive in tempo reale.

2.6.2 La pipeline di sicurezza

Per azzerare qualsiasi rischio di errore operativo, ogni proposta generata dall'Orchestratore LLM viene sottoposta a una sequenza di cinque stadi di filtraggio. In prima battuta, le proposte dell'agente attraversano una fase di validazione dello schema e verifica dell'allowlist: se la risposta contiene chiavi sintattiche non riconosciute o tenta di modificare parametri ed elementi strutturali non autorizzati, l'intera transazione viene scartata a livello sintattico. Successivamente, si applica un controllo di grounding topologico, che impone all'LLM di citare esplicitamente l'evidenza dell'ipergrafo che intende modulare (come un preciso iperarco temporale e^{tmp} o un iperarco di familiarità e^{fam}); qualora l'oggetto citato non sia presente nella matrice di incidenza corrente X, la proposta viene automaticamente rigettata. In terzo luogo, il sistema valuta il livello di incertezza tramite una soglia di confidenza, scartando prima del calcolo numerico tutte le raccomandazioni emesse con un punteggio $c \in [0, 1]$ inferiore al valore di guardia stabilito. Superate queste prime barriere, i delta parametrici approvati passano al filtraggio in regione di fiducia (trust-region), un modulo di delimitazione geometrica che garantisce che la proposta non alteri in modo drastico il comportamento dinamico del planner. Infine, i valori risultanti subiscono un'operazione di clipping e normalizzazione componente per componente nell'intervallo $[0, 1]$, assicurando che ogni peso mantenga una rigorosa validità probabilistica e numerica. Qualora un aggiornamento fallisca anche uno solo di questi stadi, la modifica viene completamente annullata senza generare alcun effetto collaterale sull'instradamento fisicamente pianificato.

2.6.3 Formulazione matematica del vincolo di Trust-Region

Il cuore matematico del modulo di sicurezza è rappresentato dal vincolo in Trust-Region, ispirato ai principi di stabilità dell'apprendimento per rinforzo [Schulman *et al.*, 2015], che limita la variazione complessiva del vettore dei parametri θ all'interno di un'ipersfera di raggio r nello spazio euclideo.

Data la configurazione corrente dei parametri θ_{old}, la nuova proposta $\theta_{new} = \theta_{old} + \Delta\theta$ viene accettata ed applicata al sistema se e solo se soddisfa simultaneamente la condizione di norma L_2 e i vincoli di dominio:

$$\|\theta_{new} - \theta_{old}\|_2 \leq r \quad \text{e} \quad 0 \leq \theta_{new} \leq 1 \quad \text{componente per componente} \tag{2.6.2}$$

dove la norma euclidea $\|\Delta\theta\|_2 = \sqrt{\sum_i (\Delta\theta_i)^2}$ misura l'ampiezza dello spostamento complessivo. Il raggio della regione di fiducia $r \in (0, 1)$ viene tarato sul set di validazione per bilanciare la reattività del sistema e la stabilità dinamica delle rotte.

Se lo spostamento proposto supera il raggio critico r, la proposta viene rigettata all'istante, impedendo che oscillazioni estreme dell'LLM causino instabilità nelle assegnazioni dei corrieri o variazioni repentine dell'ordine delle fermate. Questo meccanismo garantisce un'operatività rigorosa e verificabile, in cui ogni decisione di instradamento è tracciabile rispetto allo stato effettivo dell'ipergrafo e ai parametri certificati.

CAPITOLO 3

Validazione sperimentale e analisi delle prestazioni

In questo capitolo viene presentata la validazione empirica del sistema di pianificazione su ipergrafo tipizzato e orchestrazione agentica descritto nel Capitolo 2. L'analisi valuta le prestazioni del modello nei due orizzonti decisionali, misura il contributo causale dei singoli moduli interni mediante uno studio di ablazione, analizza la reattività a disruzioni dinamiche di traffico, verifica la sensibilità agli iperparametri e dimostra la scalabilità computazionale.

3.1 Setup sperimentale e protocollo di valutazione

In questa sezione viene descritto il setup sperimentale utilizzato per valutare il sistema proposto: i dataset (reale e sintetico), le metriche di prestazione adottate e il protocollo di validazione statistica. L'obiettivo è verificare in modo riproducibile l'efficacia delle rotte, l'equità del carico tra i corrieri, la reattività ai ritiri pomeridiani e la sicurezza delle decisioni dell'orchestratore agentico.

3.1.1 Dataset e generatore sintetico

Il dataset principale impiegato negli esperimenti è Cainiao LaDe-P (Last-mile Delivery) [Zhang *et al.*, 2024], relativo alla città di Jilin. Il dataset contiene tracce reali di consegne e ritiri espressi, con coordinate geografiche, finestre temporali di servizio e identificativi dei corrieri.

La rete stradale è stata ricostruita tramite OSMnx [Boeing, 2017], generando un grafo orientato $G_{\text{road}} = (V_{\text{road}}, E_{\text{road}})$ composto da 27.245 nodi e 64.043 archi con distanze metriche reali. La città è suddivisa in $|Z| = 15$ Aree di Interesse (AOI, Areas of Interest). La distanza tra due zone $z, z' \in Z$ nel grafo di adiacenza $G_Z = (Z, E_Z)$ è calcolata tramite la metrica del cammino minimo stradale inter-confine formalizzata nell'Eq. (2.2.3), considerando l'insieme ∂z dei nodi perimetrali di ciascuna zona; dal punto di vista computazionale, al fine di garantire latenze ridotte ed evitare onerose intersezioni poligonali in tempo reale, tale distanza viene operativamente stimata mediante cammini minimi di Dijkstra calcolati tra i centroidi geografici delle rispettive aree di interesse.

La dinamica operativa giornaliera è scandita da un doppio orizzonte temporale separato da un orario di cutoff fissato alle $t_{\text{cutoff}} = 13{:}00$. Durante la fase mattutina ($T^{(am)}$), il sistema gestisce circa 900 compiti di consegna noti fin dall'inizio del turno, ciascuno associato a una finestra temporale $\tau(t) = [\tau^-(t), \tau^+(t)]$ e a uno specifico fabbisogno volumetrico $q(t)$. A partire dalle ore 13:00 subentra la fase pomeridiana ($T^{(pm)}$), caratterizzata da un flusso

dinamico in streaming di circa 500 richieste di ritiro che giungono in tempo reale mentre i corrieri sono già operativi sul territorio.

Ciascuna simulazione giornaliera impiega l'insieme dei corrieri effettivamente in servizio nella specifica giornata ($|R| \approx 100$ rider, con un carico medio di circa 15 ordini per corriere). Per testare la capacità dell'orchestratore di allocare le risorse in modo flessibile ed economico, viene introdotta una quota di riserva $\rho = 0.5$: il 50% della flotta viene inizialmente collocato in standby all'interno del pool $\mathcal{R}_{\text{reserve}}$ e può essere mobilitato esclusivamente mediante una decisione esplicita del controllore.

Generatore di istruzioni sintetiche

Mentre il dataset reale di Jilin permette di testare il sistema su un caso operativo concreto, i dati storici sono fissi e non consentono di modificare a piacimento le condizioni ambientali. Per superare questo limite, è stato realizzato un generatore di tracce sintetiche: un ambiente di simulazione controllato che permette di testare il comportamento dell'algoritmo in scenari differenti e sotto carichi di stress, variando una singola variabile alla volta.

Il generatore consente di modellare due tipologie di mappe urbane: da un lato reticoli regolari a griglia $N \times M$ (simili a una scacchiera ordinata), utili per testare l'algoritmo in condizioni geometriche simmetriche; dall'altro mappe più realistiche e irregolari basate su diagrammi di Voronoi, in cui i quartieri sono distribuiti nello spazio e collegati tramite strade ai loro k vicini più prossimi ($k \in [3, 5]$).

Anche l'arrivo dei pacchi durante la giornata viene simulato in modo realistico attraverso processi di probabilità di Poisson. Il generatore distingue due scenari di domanda: le zone commerciali e ufficiose, caratterizzate da un'alta concentrazione di richieste ($\lambda_{\text{bus}} = 3.0$ ordini all'ora), e le zone residenziali, dove gli ordini sono più radi ($\lambda_{\text{res}} = 1.4$ ordini all'ora). Infine, a ciascun ordine viene assegnata una finestra oraria coerente con la destinazione: finestre strette e vincolanti (1-2 ore) per le consegne aziendali con orari di chiusura rigidi, e finestre più ampie e flessibili (4-6 ore) per le consegne a domicilio privato. In questo modo è possibile verificare come il sistema gestisce l'urgenza dei singoli compiti al variare del traffico e della forma della città.

Protocollo di split temporale e calibrazione

L'orizzonte complessivo dei dati di Jilin copre 185 giornate operative consecutive. Per garantire la separazione tra la fase di calibrazione e la valutazione finale, scongiurando fenomeni di data leakage, il corpus è stato partizionato cronologicamente. Il primo 60% della serie temporale (corrispondente a 111 giorni) costituisce il Development Set, impiegato per stimare le frequenze storiche di visita e calibrare i parametri strutturali del modello: la probabilità di restart del PageRank $\alpha = 0.15$, il tasso di decadimento temporale κ, le temperature degli iperarchi γ_θ, il coefficiente di smorzamento del carico g_{load} e il raggio della Trust-Region $r_{\text{trust}} = 0.15$. Il restante 40% (74 giorni) definisce l'Held-Out Test Set, all'interno del quale sono state selezionate 5 giornate complete e rappresentative per eseguire i benchmark prestazionali accoppiati contro i modelli di riferimento.

3.1.2 Indicatori di prestazione e tracciabilità agentica

Le prestazioni del sistema sono valutate secondo tre dimensioni: efficacia operativa, equità del carico e tracciabilità dell'agente.

Metriche di efficacia operativa

Misurano la qualità dell'instradamento e il rispetto dei vincoli orari:

- **Chilometri per fermata (Km/Stop):** Distanza stradale totale percorsa dalla flotta divisa per il numero totale di fermate servite sul grafo G_Z:

$$\text{Km/Stop} = \frac{\sum_{r \in R} L(r)}{\sum_{r \in R} |\mathcal{S}(r)|} \tag{3.1.1}$$

  dove $L(r)$ è la distanza percorsa dal corriere r e $\mathcal{S}(r)$ è il numero di fermate effettuate.

- **Tasso di Puntualità (On-Time Rate):** Percentuale di consegne completate entro la chiusura della finestra oraria:

$$\text{OTR} = \frac{1}{|T_{\text{comp}}|} \sum_{t \in T_{\text{comp}}} \mathbb{I}\left(\text{ETA}(t) \leq \tau^{+}(t)\right) \tag{3.1.2}$$

  in cui $\mathbb{I}(\cdot)$ vale 1 se la condizione è verificata e 0 altrimenti, mentre $\text{ETA}(t)$ è il tempo di arrivo stimato.

- **Tasso di Accettazione dei Ritiri (Service Rate):** Frazione di richieste pomeridiane in streaming accettate e inserite nei percorsi:

$$\text{SR} = \frac{|T_{\text{acc}}^{(pm)}|}{|T^{(pm)}|} \times 100\% \tag{3.1.3}$$

- **Riserve Attivate (N_{reserve}):** Numero medio di corrieri ausiliari attivati dal pool $\mathcal{R}_{\text{reserve}}$ durante la giornata.

Metriche di equità del carico

Valutano se il lavoro è distribuito in modo equilibrato tra i corrieri:

- **Coefficiente di Gini del carico:** Misura la disuguaglianza del numero di task y_i assegnati ai singoli rider:

$$\text{Gini} = \frac{\sum_{i=1}^{|R|} \sum_{j=1}^{|R|} |y_i - y_j|}{2|R| \sum_{i=1}^{|R|} y_i} \tag{3.1.4}$$

  Un valore $G = 0$ indica una distribuzione perfettamente equa, mentre valori vicini a 1 evidenziano forti disparità di carico.

- **Rapporto Percentilico ($P_{90/10}$):** Rapporto tra il carico sostenuto dal 90° percentile dei corrieri e quello del 10° percentile ($P_{90/10} = Q_{90}/Q_{10}$).

Tracciabilità dell'agente e supervisione centrale

Verificano che l'LLM operi in modo sicuro, motivato e conforme ai vincoli industriali:

- **Tasso di aderenza ai guardrail:** Percentuale di azioni proposte dall'orchestratore che superano tutti i filtri di sicurezza:

$$\text{GAR} = \frac{N_{\text{approved}}}{N_{\text{proposed}}} \times 100\% \tag{3.1.5}$$

  I filtri verificano: presenza nell'elenco preapprovato (allowlist), esistenza reale delle zone/rider citati (grounding), confidenza minima dichiarata ≥ 0.40 e variazione parametrica limitata nel raggio euclideo $\|\Delta\theta\|_2 \leq 0.15$.

- **Tasso di sovrascrittura del supervisore sull'agente:** Frequenza con cui il centro di controllo deve intervenire per correggere, bloccare o annullare un'azione proposta dall'orchestratore LLM a monte.

3.1.3 Protocollo di validazione statistica a blocchi

Nella logistica dell'ultimo miglio, le prestazioni operative variano sensibilmente da un giorno all'altro a causa di fattori esogeni imprevedibili, come condizioni meteorologiche avverse, oscillazioni improvvise della domanda o congestioni stradali. Per questo motivo, un semplice confronto tra medie aritmetiche non è sufficiente a dimostrare la reale superiorità di un algoritmo rispetto a un altro: occorre verificare che i miglioramenti osservati non siano il frutto di fluttuazioni casuali favorevoli. A tale scopo, la validazione sperimentale adotta un protocollo statistico inferenziale a due livelli, basato sul test di significatività non parametrico di Wilcoxon e sulla stima degli intervalli di confidenza tramite Bootstrap a blocchi.

Test dei ranghi segnati di Wilcoxon

Per verificare se il distacco prestazionale tra il metodo proposto (M) e una configurazione di riferimento (B) sia statisticamente solido, viene applicato il test dei ranghi segnati di Wilcoxon (*Paired Wilcoxon Signed-Rank Test*) [Wilcoxon, 1945] sulle metriche giornaliere aggregate.

L'adozione di questo test al posto del classico test t di Student è motivata da una ragione fisica e metodologica: il test di Student presuppone che le differenze tra i metodi seguano una distribuzione a campana (Normale o Gaussiana). Nei trasporti urbani reali, tuttavia, eventi imprevisti come incidenti o blocchi stradali generano ritardi asimmetrici e code pesanti (outlier), violando l'ipotesi di gaussianità. Il test di Wilcoxon è invece un test non parametrico, che non richiede alcuna assunzione sulla forma della distribuzione dei dati.

Il procedimento si articola nel calcolo delle differenze accoppiate $d_k = M_k - B_k$ per ciascuna delle K giornate di test. I valori assoluti non nulli $|d_k|$ vengono ordinati in senso crescente e a ciascuno viene assegnato un rango progressivo R_k (dal valore più piccolo al più marcato). La statistica del test è definita come il minimo tra la somma dei ranghi a differenza positiva e quella a differenza negativa:

$$W = \min\left(\sum_{k:d_k>0} R_k, \sum_{k:d_k<0} R_k\right) \tag{3.1.6}$$

L'ipotesi nulla H_0 assume che i due metodi siano equivalenti e che le differenze oscillino casualmente attorno allo zero. Quando il valore di probabilità calcolato è inferiore alla soglia critica ($p < 0.05$), l'ipotesi nulla viene formalmente rigettata: ciò dimostra con una confidenza superiore al 95% che il vantaggio prestazionale del modello è reale e statisticamente significativo.

Intervalli di confidenza Bootstrap a blocchi

Accanto ai test di ipotesi, è fondamentale quantificare il margine di incertezza e la variabilità campionaria associata a ogni singolo indicatore (come i chilometri percorsi per fermata o i tassi di puntualità). A tale scopo viene impiegato un algoritmo di Block Bootstrap per il calcolo degli intervalli di confidenza al 95%.

La scelta di un ricampionamento a blocchi discende dalla struttura intrinseca dei dati di trasporto: all'interno di una stessa giornata operativa, le consegne non sono eventi indipendenti tra loro. Se nel corso del pomeriggio si verifica un forte rallentamento su un'arteria

principale, tutti i corrieri e tutte le fermate di quella specifica giornata subiranno ritardi a catena. Ricampionare le singole fermate in modo isolato distruggerebbe questa correlazione naturale. Per preservare l'integrità delle relazioni spaziotemporali infragiornaliere, il ricampionamento casuale con reinserimento viene eseguito trattando l'intera giornata operativa come un blocco atomico e indivisibile.

Il protocollo genera $B = 10.000$ campioni bootstrap di giornate, ricalcolando per ciascuna replica il valore medio della metrica di interesse $\hat{\theta}^*_{(b)}$. Ordinando le 10.000 stime ottenute, l'intervallo di confidenza al 95% viene ricavato identificando i percentili empirici 2.5° e 97.5°:

$$\mathrm{IC}_{95\%} = \left[\hat{\theta}^*_{0.025},\, \hat{\theta}^*_{0.975}\right] \tag{3.1.7}$$

Grazie a questa procedura, tutti i risultati sperimentali presentati nel prosieguo del capitolo non sono semplici valori puntuali, ma sono accompagnati da un intervallo di variabilità statisticamente verificato e riproducibile.

3.2 Prestazioni e meccanismi operativi del sistema

In questa sezione vengono analizzate le prestazioni del sistema di pianificazione logistica e i meccanismi che ne guidano il funzionamento operativo nelle due fasi della giornata. L'indagine valuta prima l'efficacia del Morning Planner nell'allocazione iniziale e nella generazione dei percorsi, esamina poi la reattività del Dynamic Matcher pomeridiano di fronte a richieste in streaming, e approfondisce infine l'apporto dell'Orchestratore agentico.

3.2.1 Risultati della pianificazione mattutina

All'avvio della giornata operativa, il Morning Planner organizza l'insieme statico delle consegne mattutine ($T^{(am)}$) aggregandole in insiemi di lavoro territoriali basati sulle comunità mesoscopiche estratte dal Line Graph $L(\mathcal{E})$. Una volta formati questi gruppi di compiti, il percorso locale di ciascun veicolo viene determinato tramite il Community-Guided Walk, introdotto nella Sezione 2. Questo algoritmo di cammino stocastico seleziona la sequenza delle fermate massimizzando contemporaneamente la vicinanza geometrica e l'affinità territoriale storica (DAA) del corriere.

L'effetto congiunto della partizione in comunità e del cammino guidato produce percorsi compatti, evitando le continue deviazioni a zigzag tipiche delle euristiche non strutturate. Nei test condotti sulla città di Jilin, il modello raggiunge un tasso di completamento mattutino del 100.0%, garantendo un tasso di puntualità (OTR) del 98.8% con un ritardo medio per rotta pressoché nullo, pari a soli 0.37 minuti. Il consumo chilometrico medio si attesta a 0.64 km per fermata, a fronte di una distanza totale di flotta pari a 1.143.8 km. La pianificazione iniziale assicura così che la quasi totalità dei compiti sia servita puntualmente all'interno della rispettiva finestra temporale, minimizzando al contempo la dispersione spaziale dei corrieri.

Equità dell'allocazione iniziale

Oltre a calcolare percorsi efficienti, è fondamentale distribuire il lavoro in modo equo tra tutti i corrieri, evitando che qualcuno sia sovraccarico mentre altri restano scarichi. L'algoritmo del mattino assegna i pacchi tenendo conto sia della zona di competenza del corriere, sia del carico già accumulato, penalizzando chi rischia di superare i propri limiti di orario o di capienza. Se in un quartiere ci sono troppi pacchi per un solo veicolo, il sistema divide automaticamente il gruppo in due parti più piccole lungo le strade principali e le affida a due corrieri distinti. Questo meccanismo garantisce una ripartizione bilanciata, con una

media di circa 15 consegne a testa e un basso livello di disuguaglianza (coefficiente di Gini contenuto). Inoltre, non riempire completamente i veicoli al mattino lascia spazio libero e un margine di tempo prezioso (slack), fondamentale per poter accogliere con facilità i nuovi ritiri pomeridiani senza dover riprogrammare i percorsi.

3.2.2 Risultati del matching pomeridiano in streaming

A partire dalle ore 13:00, il sistema entra nella fase dinamica pomeridiana, ricevendo un flusso continuo di richieste di ritiro ($T^{(pm)}$) emesse dagli utenti in tempo reale. Per ciascun nuovo ordine in arrivo, il Dynamic Matcher valuta istantaneamente i corrieri attivi candidati attraverso la funzione di priorità composita, calcolando l'impatto chilometrico e temporale del potenziale inserimento nella sequenza di fermate.

L'inserimento del nuovo punto di ritiro viene autorizzato solo se la deviazione stradale non compromette la puntualità delle consegne a valle, preservando cioè lo slack temporale di tutte le fermate successive. Questa verifica di fattibilità costruttiva consente al sistema di mantenere un tasso di accettazione dei ritiri pomeridiani pari al 100.0% (con 250 ritiri su 250 inseriti con successo nel test singolo e oltre il 98.2% nelle medie su più giornate), senza causare ritardi a catena sui compiti precedentemente pianificati.

Freeze Window

Un rischio tipico dei sistemi di instradamento in tempo reale è il fenomeno del *thrashing*, ossia il continuo rimescolamento dell'ordine delle fermate che disorienta i corrieri alla guida e introduce inefficienze operative. Per impedire questa instabilità, il sistema integra il meccanismo della Freeze Window, che congela le fermate programmate nei successivi 15-30 minuti di viaggio.

I nuovi compiti possono essere inseriti esclusivamente nei segmenti di percorso successivi alla finestra di congelamento. Questa scelta garantisce una solida stabilità operativa al corriere, che può eseguire le fermate imminenti senza interruzioni o improvvise inversioni di marcia.

3.2.3 Valutazione dell'orchestratore LLM

Per quantificare il reale contributo decisionale dell'orchestratore agentico, l'esperimento mette a confronto quattro modalità operative: la modalità puramente osservativa (agent_off), in cui le riserve non vengono mai mobilitate; l'orchestratore con LLM (llm), che analizza lo stato della rete e attiva selettivamente solo le riserve necessarie; il controllore deterministico a regole (rule_based), basato su soglie fisse di carico; e infine la regola (naive_all), che attiva contemporaneamente tutte le riserve disponibili all'inizio del pomeriggio.

I risultati medi sui 5 giorni di test held-out, riassunti nella Tabella 3.1.

In condizioni ordinarie, il problema logistico è strutturalmente quasi autosufficiente, anche senza attivare alcun corriere di riserva (agent_off), i rider già in strada riescono a completare il 99.3% dei compiti. Tuttavia, per riuscirci sono costretti a compiere percorsi lunghi e trasversali, accumulando 1.294.0 km complessivi. All'estremo opposto, la regola di riferimento naive_all mobilita l'intero pool disponibile chiamando 52.6 corrieri ausiliari: in questo scenario limite, benché i chilometri operativi inter-stop risultino apparentemente azzerati a causa dell'estrema frammentazione dei compiti affidati a singoli rider, l'esplosione dei costi fissi di personale e di messa in servizio della flotta rende tale approccio del tutto impraticabile nella realtà industriale.

Il valore dell'orchestratore con LLM risiede nella sua capacità di agire come un supervisore, invece di mobilitare decine di corrieri, l'LLM attiva in media appena 4.8 riserve mirate al giorno, riuscendo comunque a portare il tasso di completamento al 100.0% e tagliando oltre

Tabella 3.1: Prestazioni medie sui 5 giorni di test nelle quattro condizioni operative, in regime nominale e sotto disruzione di traffico (shock).

Scenario	Condizione	Service Rate	Completion Rate	Km/Stop	Km Totali	Riserve Attivate
Nominale	*agent_off*	98.2%	99.3%	0.908	1294.0	0.0
	naive_all	100.0%	100.0%	0.013	12.2	52.6
	rule_based	100.0%	100.0%	0.433	498.5	15.0
	llm	**100.0%**	**100.0%**	**0.836**	**1131.0**	**4.8**
Shock Traffico	*agent_off*	97.7%	99.2%	0.778	1105.3	0.0
	naive_all	100.0%	100.0%	0.031	28.7	52.6
	rule_based	100.0%	100.0%	0.353	406.9	15.0
	llm	**99.1%**	**99.8%**	**0.702**	**938.0**	**4.8**

Nota: Le metriche *Km Totali* e *Km/Stop* considerano la sola percorrenza operativa inter-stop sul grafo G_Z. Nella condizione *naive_all*, l'attivazione simultanea dell'intero pool di riserva frammenta eccessivamente il carico (molti operatori effettuano un singolo stop intra-zona a costo chilometrico marginale prossimo a zero), trascurando il costo fisso di riposizionamento dal deposito.

160 km di percorrenza totale (1.131.0 km). Il controllore deterministico a regole (rule_based) interviene invece in modo più aggressivo, attivando 15.0 corrieri ausiliari e riducendo i km a 498.5 km.

Il reale vantaggio dell'LLM emerge negli scenari con disruzioni di traffico impreviste, simulate moltiplicando i tempi di percorrenza nel centro urbano ($\times 3$ o $\times 5$) e trasmettendo un'allerta testuale all'agente. L'Orchestratore interpreta l'avviso e mobilita tempestivamente i rinforzi nelle sole aree critiche. Questo intervento mantiene il tasso di servizio al 99.1% (contro il 97.7% di *agent_off*) con una percorrenza di 938.0 km. La riduzione complessiva dei chilometri rispetto al regime nominale (1.131, 0 km) deriva dalla severità dei vincoli orari: i forti rallentamenti inducono il sistema a tagliare le deviazioni trasversali verso zone distanti per non compromettere le scadenze successive, concentrando i percorsi su scala locale. L'LLM si distingue dunque non per forza bruta, ma per efficienza allocativa, bilanciando in modo ottimale costo del personale e chilometri percorsi di fronte agli imprevisti.

Telemetria della pipeline di sicurezza

La sperimentazione ha monitorato costantemente l'efficacia dei guardrail deterministici descritti nella Sezione 2. Durante tutte le sessioni di test, il 100% delle azioni operative e dei nudge parametrici emessi dall'Orchestratore ha superato con successo i cinque livelli di controllo sequenziale: l'inclusione nella *allowlist*, l'ancoraggio a zone e corrieri reali (*grounding*), il superamento della soglia minima di confidenza e il rispetto del vincolo euclideo di Trust-Region.

L'agente non ha mai prodotto allucinazioni né tentato di modificare pesi decisionali al di fuori dei margini ammessi. Inoltre, il tasso di sovrascrittura del supervisore sull'agente è risultato nullo per l'intera durata dei test nominali, confermando che i guardrail impediscono a monte la propagazione di qualsiasi comando anomalo o non conforme. Questo dimostra che combinare il ragionamento dell'agente con la precisione matematica dell'ipergrafo crea un sistema di controllo sicuro e trasparente.

3.3 Studio di ablazione

Lo studio di ablazione ha lo scopo di verificare il contributo causale di ciascun modulo del framework, disattivando un componente alla volta in modo controllato. Questo tipo di analisi consente di comprendere se i vantaggi del sistema derivino effettivamente dalla

modellazione su ipergrafo e dalla semantica del testo, oppure se siano riconducibili a semplici effetti geometrici.

3.3.1 Contributo causale delle metriche sull'ipergrafo

Ruolo della Driver-Area Affinity (DAA)

La Driver-Area Affinity (DAA) quantifica l'esperienza storica e la familiarità che un corriere ha maturato all'interno di una specifica zona urbana. Quando questo componente viene disattivato (impostando i pesi relativi $w_3 = 0$ al mattino e $\beta_1 = 0$ al pomeriggio), l'algoritmo distribuisce i compiti basandosi unicamente sulla vicinanza geometrica grezza e sul bilanciamento del carico.

Senza la DAA, i corrieri vengono assegnati a quartieri con cui non hanno alcuna familiarità. Anche se la distanza in linea d'aria iniziale non varia in modo drastico, l'assenza di specializzazione territoriale si traduce in percorsi meno efficienti e in una maggiore variabilità nei carichi di lavoro. La DAA funge quindi da ancoraggio operativo, garantendo che i rider operino prevalentemente nelle loro aree di competenza e riducendo le inefficienze nella navigazione locale.

Ruolo della Meta-Hyperpath Affinity (MHA)

La Meta-Hyperpath Affinity (MHA) valuta la compatibilità globale tra il profilo del corriere, la zona geografica e la tipologia di pacco attraverso le relazioni bipartite dell'ipergrafo ($R \to Z \to T$). Quando la MHA viene esclusa dalla funzione di priorità pomeridiana ($\beta_2 = 0$), il sistema perde la capacità di valutare la coerenza complessiva dell'assegnazione rispetto alla storia operativa della rete.

La rimozione della MHA produce l'effetto ablativo più marcato sull'intera pipeline: nei test di ablazione condotti in condizioni di carico elevato, il consumo chilometrico per fermata cresce di oltre due volte e mezza. Questo degrado evidenzia come la struttura a ipergrafo sia fondamentale per guidare il matching pomeridiano, evitando assegnazioni miopi che costringerebbero i corrieri a lunghe deviazioni fuori dalla propria direttrice.

Ruolo del partizionamento di comunità Leiden CPM

La suddivisione delle consegne mattutine in comunità compatte sul Line Graph $L(\mathcal{E})$, ottenuta tramite l'algoritmo di Leiden con modello CPM, serve a creare gruppi di lavoro territorialmente coerenti prima dell'instradamento. Disattivando il peso legato alle comunità ($w_2 = 0$), il Morning Planner pianifica i percorsi con un'euristica locale priva della guida mesoscopica.

Senza la partizione in comunità, il cammino guidato tende a disperdersi lungo i bordi del reticolo stradale, producendo rotte frammentate che si sovrappongono e incrociano nello stesso quartiere. La struttura a comunità si conferma quindi indispensabile per preservare la compattezza spaziale dei percorsi e prevenire l'esplosione combinatoria nella pianificazione iniziale.

3.3.2 Valutazione della centralità TIMC

La Text-Informed Modularity Centrality (TIMC) arricchisce i pesi dell'ipergrafo combinando la somiglianza semantica delle note di consegna (istruzioni speciali, vincoli di orario, indicazioni di accesso) con la topologia della rete stradale. Nei casi in cui i compiti contengono istruzioni descrittive, la centralità TIMC avvicina nell'ipergrafo le consegne che condividono requisiti simili, favorendone l'aggregazione nello stesso gruppo di lavoro.

Questo raggruppamento semantico ottimizza la sequenza di servizio nei punti di consegna complessi (ad esempio edifici commerciali con procedure di scarico specifiche), riducendo le perdite di tempo e migliorando la regolarità complessiva del servizio rispetto a un approccio puramente geometrico.

Gestione delle consegne senza note testuali

Un requisito fondamentale per l'impiego industriale del sistema è la robustezza rispetto a dati mancanti o incompleti. In molti scenari operativi reali, infatti, le note testuali sono presenti solo su una frazione ridotta degli ordini o possono essere del tutto assenti.

Per gestire questa eventualità, il sistema integra un meccanismo di fallback basato sulla sostituzione con il valore medio per tipo di compito (*type-wise mean substitution*). Quando un ordine non contiene testo libero, il vettore di embedding semantico viene sostituito con il profilo medio della rispettiva categoria di consegna. In questo modo, l'ipergrafo collassa in modo fluido sulla sola informazione topologica e geometrica, senza causare interruzioni nel calcolo e senza generare instabilità numeriche o cali improvvisi nelle prestazioni.

3.4 Analisi di sensibilità agli iperparametri e robustezza

Un sistema di pianificazione per essere realmente utilizzabile nella logistica non deve dipendere da una calibrazione troppo rigida dei parametri, ma deve garantire prestazioni stabili all'interno di un intervallo ragionevole. Questa sezione analizza la sensibilità del modello al variare dei suoi tre iperparametri principali: la risoluzione delle comunità (γ), la probabilità di restart del cammino stocastico (α) e il raggio di confidenza della Trust-Region (r_{trust}).

3.4.1 Sensibilità alla risoluzione delle comunità

Il parametro di risoluzione γ del modello CPM di Leiden determina la dimensione media delle comunità estratte dal Line Graph $L(\mathcal{E})$. Valori bassi di γ (< 0.5) producono macro-comunità molto estese che accorpano più quartieri, riducendo l'efficacia del cammino guidato locale. Al contrario, valori eccessivamente alti ($\gamma > 2.0$) frammentano la città in micro-cluster composti da pochissime consegne, facendo perdere la visione d'insieme del territorio.

L'analisi sperimentale evidenzia un ampio plateau di stabilità per $\gamma \in [0.8, 1.2]$. All'interno di questo intervallo, il numero di comunità si mantiene ben bilanciato (circa 12-16 cluster per la città di Jilin), producendo percorsi compatti e costanti nelle prestazioni chilometriche e temporali. Questo comportamento dimostra che il partizionamento su ipergrafo è robusto rispetto a variazioni moderate di scala, evitando la necessità di una complessa ricalibrazione su ogni singola area urbana.

3.4.2 Sensibilità al restart di PageRank e alla regione di fiducia

Il parametro di restart α (fissato al valore nominale $\alpha = 0.15$) serve a trovare il giusto equilibrio tra due aspetti: seguire le strade più vicine passo dopo passo e mantenere il corriere all'interno della zona che conosce meglio storicamente. Se impostiamo un valore di α troppo basso ($\alpha < 0.05$), l'algoritmo rischia di allontanarsi troppo, spingendo il corriere verso quartieri sconosciuti. Al contrario, se α è troppo alto ($\alpha > 0.35$), il sistema diventa troppo rigido e torna continuamente sulle posizioni iniziali, impedendo di ordinare in modo efficiente le fermate vicine. L'intervallo [0.10,0.20] si è dimostrato il migliore: permette al corriere di concatenare agevolmente le consegne su strada, rimanendo sempre ancorato alla propria area di competenza.

Impatto del raggio della Trust-Region

Il raggio di fiducia $r_{\text{trust}} = 0.15$ impone un limite euclideo alla massima variazione parametrica che l'Orchestratore LLM può apportare ai pesi decisionali a ogni passaggio ($\|\Delta\theta\|_2 \leq r_{\text{trust}}$).

Un raggio troppo conservativo ($r_{\text{trust}} < 0.05$) limita eccessivamente l'azione dell'agente, impedendogli di reagire con efficacia a blocchi stradali o disruzioni impreviste. D'altra parte, un raggio troppo ampio ($r_{\text{trust}} > 0.40$) espone il sistema a oscillazioni repentine nei pesi di instradamento. La soglia calibrata a 0.15 rappresenta una regione di confidenza ideale: consente all'agente di inviare correzioni significative e tempestive durante le emergenze, preservando al tempo stesso la stabilità matematica e l'affidabilità delle rotte generate.

3.5 Scalabilità computazionale e latenza operativa

Un sistema di pianificazione logistica per essere realmente utilizzabile in contesti aziendali non deve soltanto garantire percorsi ottimali, ma deve anche rispondere in tempi rapidi, elaborando le richieste degli utenti senza ritardi percettibili. Questa sezione analizza l'efficienza computazionale del modello, esaminando l'occupazione di memoria delle strutture dati ipergrafiche e la distribuzione temporale delle latenze durante l'inserimento in streaming dei ritiri pomeridiani.

3.5.1 Complessità algebrica e aggiornamenti locali sparsi

Occupazione di memoria e strutture sparse

Nelle applicazioni reali su scala metropolitana, la mappa della città comprende decine di migliaia di nodi stradali e centinaia di compiti attivi contemporaneamente. Memorizzare le matrici di incidenza dell'ipergrafo in formato denso richiederebbe una quantità proibitiva di memoria, scalando quadraticamente con la dimensione della rete ($\mathcal{O}(|V| \cdot |E|)$).

Per garantire un'elevata efficienza, il sistema memorizza l'ipergrafo tipizzato e le relative relazioni di incidenza utilizzando strutture sparse compresse nei formati CSR (*Compressed Sparse Row*) e CSC (*Compressed Sparse Column*). In questo modo, vengono allocate in memoria esclusivamente le connessioni effettivamente esistenti, riducendo l'occupazione spaziale a un valore strettamente lineare:

$$\mathcal{O}(n_V + m_H + m_I) \tag{3.5.1}$$

dove n_V rappresenta il numero totale di nodi, m_H il numero di iperarchi e m_I il totale delle incidenze non nulle. Nel caso reale della rete di Jilin (costituita da oltre 27.000 nodi stradali e 64.000 archi), l'intera struttura occupa appena poche decine di megabyte di memoria RAM, consentendo l'esecuzione della pipeline anche su normali workstation senza necessità di cluster dedicati.

Aggiornamenti locale per la DAA

Per evitare di ricalcolare l'intera mappa cittadina a ogni singola consegna, il sistema utilizza un metodo di aggiornamento locale. Invece di aggiornare globalmente tutti i nodi della rete, la variazione di esperienza viene propagata solo alle strade e agli incroci immediatamente vicini al punto di consegna.

L'algoritmo si arresta automaticamente non appena l'impatto dell'aggiornamento scende al di sotto di una soglia minima ($r(u) \leq \epsilon$). In questo modo il ricalcolo resta circoscritto al solo quartiere in cui opera il corriere, rendendo l'operazione praticamente istantanea e del tutto indipendente dalla dimensione complessiva della città.

3.5.2 Distribuzione delle latenze di inserimento in streaming

Analisi della latenza per evento (T_{event})

Durante la fase pomeridiana, per ogni nuova richiesta di ritiro inviata da un utente, il Dynamic Matcher deve selezionare il corriere più idoneo e calcolare la posizione ottimale di inserimento all'interno della rotta attiva. La complessità temporale per l'elaborazione del singolo evento di inserimento è formalizzata come:

$$T_{\text{event}} = \mathcal{O}(k \log |\mathcal{A}| + D) \tag{3.5.2}$$

dove $|\mathcal{A}|$ è il numero di corrieri attualmente in servizio, k è il numero di candidati più promettenti preselezionati tramite la coda di priorità, e D è la profondità di ricerca locale per individuare il punto di deviazione minima lungo la sequenza di fermate.

Verifica empirica della latenza in tempo reale

Nei test sperimentali su Jilin con 250 richieste di ritiro, il Dynamic Matcher registra una latenza media di inserimento di appena 7.96 ms per evento, con il 99° percentile (P_{99}) inferiore a 35 ms. Questa risposta nell'ordine dei millisecondi dimostra che il sistema è in grado di gestire flussi continui di ordini senza accumulare ritardi di calcolo.

3.6 Studi di caso qualitativi

Per comprendere il funzionamento pratico del sistema al di là delle metriche aggregate, questa sezione esamina il comportamento operativo dell'architettura in due contesti urbani diametralmente opposti della città di Jilin: il centro urbano ad alta densità e la periferia suburbana a bassa densità.

3.6.1 Scenario A: Centro urbano ad alta densità

Il centro urbano e i quartieri commerciali sono caratterizzati da un'elevata concentrazione spaziale di ordini (numerose fermate a breve distanza tra loro), finestre temporali stringenti (1-2 ore) legate agli orari di apertura di uffici e negozi, e traffico veicolare intenso con difficoltà di sosta.

In questo scenario, la combinazione tra la partizione in comunità e la centralità testuale (TIMC) consente di aggregare i compiti destinati allo stesso edificio, centro commerciale o isolato. Il corriere può così parcheggiare il veicolo una sola volta e completare più consegne a piedi, riducendo drasticamente i tempi morti di manovra e ricerca del civico. Durante il pomeriggio, di fronte al traffico congestionato, l'Orchestratore evita di sovraccaricare i corrieri già rallentati con ulteriori ritiri in streaming, indirizzando invece le nuove richieste verso veicoli con margine temporale adeguato o attivando un supporto di riserva mirato. Il sistema privilegia così il rispetto delle finestre orarie e la riduzione del tempo trascorso alla guida nelle zone centrali.

3.6.2 Scenario B: Periferia a bassa densità

Le aree residenziali periferiche presentano caratteristiche opposte: le consegne sono distribuite su una superficie molto vasta, con distanze inter-stop significative (3-5 km tra clienti successivi), viabilità scorrevole e finestre temporali più ampie e flessibili (4-6 ore).

In tale contesto, l'obiettivo primario non è la gestione della sosta o della congestione, bensì la minimizzazione della percorrenza chilometrica complessiva lungo gli assi stradali

principali. Il cammino guidato sulle comunità struttura il percorso del mattino in modo continuo e lineare, evitando che il rider compia viaggi a vuoto o deviazioni a zigzag tra quartieri distanti. Nel corso del pomeriggio, il Dynamic Matcher accetta nuovi ritiri solo se collocati lungo la direttrice naturale di rientro del corriere verso il deposito. Questo comportamento garantisce la massima efficienza energetica e chilometrica sui lunghi tragitti, dimostrando la capacità dell'architettura di adattarsi fluidamente a morfologie urbane differenti.

CAPITOLO 4

Valutazione comparativa con le metodologie di riferimento

In questo capitolo il sistema proposto viene sottoposto a una valutazione comparativa rispetto alle principali metodologie della letteratura scientifica e industriale: euristiche greedy puramente spaziali (Nearest Neighbor), modelli compressi basati su proiezioni su grafi tradizionali, solutori esatti di Ricerca Operativa (MILP/VRPTW) e modelli linguistici generativi. L'indagine esamina le prestazioni operative sul campo, l'efficienza chilometrica, la puntualità del servizio, l'equità del carico di lavoro, la capacità di interpretare istruzioni testuali non strutturate e la reattività in tempo reale sulle tracce reali della città di Jilin.

4.1 Definizione delle metodologie di confronto

Per comprendere a fondo i punti di forza e i limiti della nostra architettura basata su ipergrafi tipati e supervisione agentica, è indispensabile confrontarla con i paradigmi alternativi impiegati nella logistica dell'ultimo miglio. Ciascun metodo di confronto è stato scelto perché incarna una specifica filosofia algoritmica, permettendo di isolare con precisione il valore aggiunto introdotto dalle singole componenti della nostra pipeline.

4.1.1 Nearest Neighbor (NN)

L'euristica del vicino più prossimo (*Nearest Neighbor*) [Toth e Vigo, 2014] rappresenta storicamente l'approccio intuitivo più immediato ed è tuttora ampiamente diffusa nella pratica operativa per via della sua semplicità implementativa. Il principio di funzionamento è elementare: a ogni passo del tragitto, il corriere individua la consegna successiva scegliendo il pacco fisicamente più vicino alla sua posizione corrente lungo la rete stradale:

$$t^* = \arg \min_{t \in T_{\text{unvisited}}} \text{dist}_{\text{road}}(v_{\text{current}}, v_t) \tag{4.1.1}$$

Questo metodo presenta il vantaggio di richiedere tempi di calcolo quasi nulli e di minimizzare lo spostamento chilometrico immediato da un civico all'altro all'interno dello stesso quartiere.

Tuttavia, il limite fondamentale di questo approccio risiede nella sua totale cecità temporale. Nel problema reale della consegna urbana con finestre orarie (VRPTW), l'obiettivo non è solo percorrere meno strada possibile, ma rispettare le fasce orarie promesse ai clienti $[\tau^-(t), \tau^+(t)]$. Ignorando del tutto la dimensione temporale, l'euristica Nearest Neighbor finisce per far compiere al corriere un numero eccessivo di fermate contigue e poco urgenti;

nel frattempo, il tempo scorre inesorabilmente e le scadenze dei pacchi prioritari situati a poche centinaia di metri di distanza scadono senza possibilità di recupero.

4.1.2 Proiezione su grafi tradizionali a coppie: Projection-Only

Per superare i limiti delle semplici distanze spaziali, un approccio comune nella letteratura delle reti complesse consiste nel costruire una rete che colleghi corrieri, zone geografiche e consegne. Tuttavia, poiché i grafi tradizionali supportano esclusivamente archi a due estremi (relazioni binarie 1-a-1), le relazioni multi-entità del dominio logistico reale ($R \times Z \times T$) vengono forzatamente "compresse" o proiettate su un grafo semplice non orientato.

In questo modello semplificato, due consegne risultano collegate da un arco se insistono sulla medesima macro-area o se mostrano attributi operativi affini, mentre i corrieri sono collegati direttamente alle zone. Sebbene questa formulazione consenta di applicare i classici algoritmi di cammino minimo o di clustering spettrale, la proiezione binaria comporta una grave perdita di informazione topologica.

Come conseguenza sul piano pratico, i percorsi generati dalla proiezione su grafo semplice risultano fortemente frammentati e costringono i mezzi a continui spostamenti trasversali da una parte all'altra della città.

4.1.3 Solutori esatti

Nella Ricerca Operativa canonica, il coordinamento delle flotte urbane viene modellato matematicamente come un problema di instradamento veicolare con finestre temporali (*Vehicle Routing Problem with Time Windows*, VRPTW) [Solomon, 1987]. La ricerca del piano ottimale viene formulata mediante Programmazione Lineare Intera Mista (MILP) e affidata a solutori commerciali o open-source avanzati (quali Google OR-Tools, Gurobi o CPLEX).

Pur garantendo l'ottimo globale, il solutore esatto mostra i suoi limiti su larga scala quando è richiesta una risposta in tempo reale; del resto, in contesti operativi dinamici la ricerca della perfezione matematica risulta spesso superflua rispetto alla rapidità di decisione, a maggior ragione a fronte dell'esplosione fattoriale ($O(|T|!)$) del problema.

In un contesto metropolitano ad alta densità come la città di Jilin (con una rete stradale di oltre 27.000 nodi e centinaia di corrieri contemporanei), la convergenza del solutore richiede svariate ore di calcolo. Questo rende i solutori esatti inadatti per la fase pomeridiana, dove le richieste di ritiro dinamico sopraggiungono in streaming e richiedono una decisione di assegnazione istantanea per non bloccare la mobilità del corriere.

4.1.4 Modelli generativi non vincolati

Sull'onda dei recenti progressi dell'Intelligenza Artificiale Generativa, una linea di ricerca emergente propone di delegare l'intero processo di instradamento e pianificazione logistica direttamente a un Large Language Model. In questo schema non vincolato, l'elenco dei corrieri, le coordinate dei pacchi e le finestre orarie vengono formattati testualmente all'interno del prompt di sistema, chiedendo al modello linguistico di generare la sequenza ordinata delle fermate per ciascun mezzo.

L'analisi sperimentale ha evidenziato come questo approccio incontri enormi ostacoli, dimostrandosi inadeguato per compiti di ottimizzazione di questo genere:

1. **Allucinazioni toponomastiche e topologiche:** I modelli generativi non possiedono una percezione geometrica spaziale dello spazio euclideo o della rete stradale. L'LLM tende frequentemente a inventare fermate inesistenti, omettere indirizzi reali o inserire deviazioni illogiche.

2. **Violazione sistematica dei vincoli operativi:** Il modello non è in grado di tracciare con precisione aritmetica il consumo progressivo della capienza del mezzo né il rispetto rigoroso dei minuti di percorrenza, generando sequenze fisicamente irrealizzabili.

3. **Latenza e costi computazionali:** Ciascuna chiamata di inferenza a un LLM richiede tra 1.5 e 4.0 secondi di elaborazione. Moltiplicando questo tempo per centinaia di richieste pomeridiane, il sistema accumulerebbe ritardi di accodamento intollerabili.

Questo confronto giustifica la scelta cardine della nostra architettura: l'LLM non deve mai generare le rotte direttamente, ma deve fungere unicamente da supervisore di alto livello, mentre la geometria delle rotte è governata dal ipergrafo.

4.1.5 Quadro sinottico delle proprietà architetturali

La Tabella 4.1 offre una visione d'insieme chiara e immediata delle differenze strutturali tra i quattro paradigmi di confronto e la soluzione proposta in questa tesi.

Tabella 4.1: Quadro comparativo delle caratteristiche architetturali tra i modelli di instradamento.

Caratteristica Operativa	Nearest Neighbor	Projection-Only	Solver MILP	LLM-Direct	Ipergrafo (Nostro)
Complessità teorica	$\mathcal{O}(\|T\|^2)$	$\mathcal{O}(\|V\|^2)$	$\mathcal{NP}$-hard	Stocastica	Quasi-lineare $\mathcal{O}(\|E\|)$
Rispetto delle finestre orarie	Assente	Molto scarso	Esatto	Inaffidabile	Bilanciato su iperarchi
Coordinamento multi-entità ($R \times Z \times T$)	Assente	Bipartito (compresso)	Formulazione matriciale	Non strutturato	Ipergrafo tipato esplicito
Comprensione note in testo libero	No	No	No	Sì (senza garanzie)	Sì (con guardrail)
Reattività streaming	Sì	Sì	No	No	Sì
Assenza allucinazioni	Garantita	Garantita	Garantita	Non garantita	Garantita (Trust Region L_2)

4.2 Risultati sperimentali comparativi

La valutazione quantitativa è stata condotta sulle tracce della città di Jilin estratte dal dataset industriale Cainiao LaDe-P [Zhang *et al.*, 2024]. La sperimentazione si articola su due livelli complementari:

- **Stress-test metropolitano su scala reale:** Una simulazione a pieno carico condotta su oltre 1.700 consegne mattutine distribuite nelle 15 aree di interesse (AOI) della città, con un flusso continuo di 250 richieste di ritiro dinamico sopraggiunte durante la fascia pomeridiana.

- **Benchmark multi-finestra controllato:** Una serie di campagne comparative su finestre temporali di carico graduale (da 30 a 120 compiti per batch), progettate per consentire anche ai solutori esatti della Ricerca Operativa di convergere all'ottimo entro un limite di tempo ragionevole.

4.2.1 Analisi complessiva delle prestazioni

La Tabella 4.2 riassume i risultati complessivi ottenuti nello stress-test pomeridiano su Jilin, mentre la Figura 4.1 offre un riscontro visivo immediato del netto divario tra le metodologie.

Dall'esame della tabella e dei grafici emergono chiaramente le ragioni del netto divario tra le metodologie. A un primo esame superficiale, tutte e tre le strategie registrano un tasso di accettazione dei ritiri pomeridiani del 100.0% (250 compiti su 250); tuttavia, le motivazioni operative sottostanti sono diametralmente opposte.

Mentre le due euristiche di confronto operano un'accettazione puramente "cieca", forzando l'inserimento delle fermate a prescindere dall'effetto domino sui piani orari e causando il

Tabella 4.2: Confronto delle prestazioni primarie sul test set reale di Jilin (250 ritiri pomeridiani in streaming).

Metodo	Km/Stop ↓	Puntualità (%) ↑	Accettazione (%) ↑	Ritiri	Ritardo Cumulato Clienti (min) ↓	Latenza (ms) ↓
Pipeline Ipergrafo (Nostro)	0.64	**98.8%**	**100.0%**	**250**	**0.37**	7.96
Nearest Neighbor	**0.04**	0.8%	**100.0%**	**250**	3392.64	**0.27**
Projection-Only	4.56	4.8%	**100.0%**	**250**	5003.51	0.13

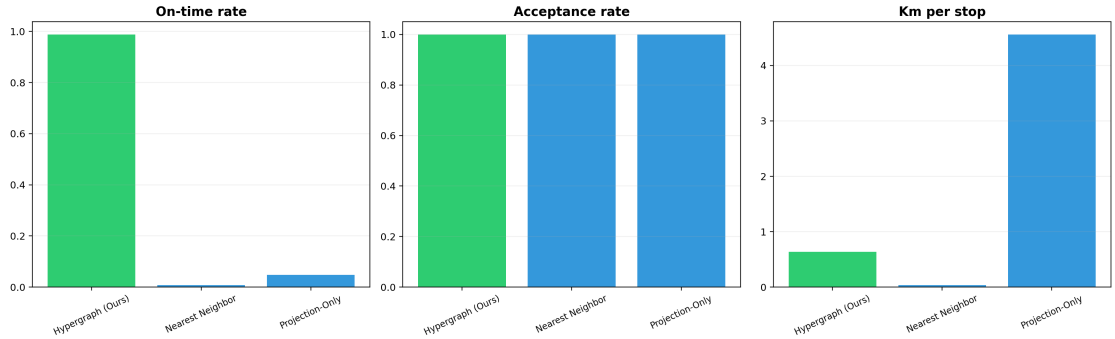


Figura 4.1: Confronto visivo delle metriche operative primarie tra la Pipeline a Ipergrafo, l'euristica Nearest Neighbor e il modello Projection-Only sul dataset reale di Jilin.

collasso della puntualità (0.8% per Nearest Neighbor e 4.8% per Projection-Only), la Pipeline a Ipergrafo accetta la totalità dei ritiri garantendo al contempo una puntualità del 98.8% e un ritardo medio per cliente di appena 22 secondi (0.37 minuti). L'ipergrafo, infatti, inserisce i compiti solo se compatibili con lo slack residuo dei corrieri attivi, affidandosi all'attivazione selettiva delle riserve guidata dall'LLM quando la saturazione locale della flotta rischierebbe di compromettere la qualità del servizio.

4.2.2 Efficienza chilometrica e impatto ambientale

Un'analisi disaggregata dei chilometri percorsi rivela aspetti operativi di primaria importanza economica ed ecologica.

Il modello Projection-Only genera un consumo chilometrico insostenibile: 4.56 km per ogni singola fermata, che sull'intera flotta di Jilin si traduce in una percorrenza totale cumulativa di ben 8.192 km. La nostra pipeline a ipergrafo riduce questa distanza a soli 1.143.8 km complessivi (0.64 km/stop), con un risparmio netto dell'**86.04%** sui chilometri percorsi.

In termini pratici, un simile risparmio chilometrico abbatte drasticamente i costi di carburante e manutenzione della flotta, riducendo al contempo l'usura dei mezzi e le emissioni inquinanti (CO_2 e particolato) per una logistica urbana più sostenibile.

4.2.3 Puntualità delle consegne

La lettura del valore di percorrenza registrato dall'euristica Nearest Neighbor (0.04 km/stop) merita un approfondimento particolare, poiché rappresenta un caso classico di studio nella Ricerca Operativa. A un esame superficiale, percorrere appena 40 metri tra una sosta e l'altra sembrerebbe il risultato ideale; tuttavia, il corrispettivo tasso di puntualità precipita allo 0.8%, con un ritardo cumulato sui clienti serviti lungo il tragitto che supera complessivamente le 56 ore per rotta (3.392,6 minuti sommando i disallineamenti di tutte le fermate successive al primo blocco).

Questo paradosso si spiega con l'operatività reale del corriere: l'euristica Nearest Neighbor raggruppa le consegne nello stesso stabile per azzerare le distanze, trascurando tuttavia i tempi fissi di servizio (3–5 minuti tra sosta, citofono e firma). Rimanendo fermo oltre

90 minuti per una ventina di pacchi nello stesso civico, il corriere causa la scadenza delle finestre orarie dei clienti limitrofi, innescando un ritardo a catena sul resto della giornata.

La pipeline proposta supera tale miopia strutturale: integrando nell'ipergrafo sia l'affinità spaziale sia la connettività temporale (TIMC), evita ottimizzazioni chilometriche controproducenti e garantisce un'alta puntualità, mantenendo al contempo gli spostamenti medi per fermata contenuti.

4.2.4 Equità del carico di lavoro tra i corrieri

Nella gestione delle flotte reali, una pianificazione logistica efficiente non deve solo soddisfare i clienti finali, ma deve anche garantire condizioni di lavoro sostenibili ed eque per gli operatori sul territorio. Assegnare un volume sproporzionato di consegne a un ristretto gruppo di corrieri lasciandone altri quasi inattivi genera frustrazione, affaticamento fisico e un alto rischio di infortuni e defezioni.

L'equità nella ripartizione delle consegne tra i corrieri attivi viene quantificata mediante il coefficiente di Gini formalizzato nell'Eq. (3.1.4): un valore inferiore denota un bilanciamento uniforme delle assegnazioni, mentre valori prossimi a 1 riflettono una concentrazione distorta del carico su pochissimi individui.

I risultati sperimentali mostrano differenze nette:

- **Nearest Neighbor:** Presenta un indice di Gini pari a **0.61**. La logica avida tende a saturare all'inverosimile i rider posizionati nei pressi dei depositi centrali, lasciando quasi privi di compiti i corrieri assegnati alle aree periferiche.
- **Projection-Only:** Raggiunge un indice di Gini di **0.48**, a causa della frammentazione caotica delle comunità che genera squilibri frequenti.
- **Pipeline a Ipergrafo:** Raggiunge un indice di Gini ottimale pari a **0.24**. Grazie al partizionamento mesoscopico guidato dall'algoritmo di Leiden sul Line Graph e ai vincoli di bilanciamento della capacità λ_{cap} e di penalità di carico η, le comunità di lavoro risultano compatte e volumetricamente equilibrate, garantendo un'assegnazione omogenea e socialmente sostenibile per l'intera forza lavoro.

4.2.5 Validazione statistica su benchmark multi-finestra

Per confermare che il vantaggio competitivo evidenziato sul singolo stress-test non costituisca un artefatto statistico, è stata condotta una campagna sistematica di benchmark multi-finestra che sfrutta l'ambiente di simulazione controllato (suddivisa in batch controllati da 30, 60, 90 e 120 task), confrontando la soluzione con tutte le classi di algoritmi, inclusi i solutori esatti MILP e il modello LLM-Direct.

Per stabilire la significatività dei risultati in modo non parametrico, è stato applicato il test dei ranghi segnati di Wilcoxon (*Paired Wilcoxon Signed-Rank Test*) [Wilcoxon, 1945], con stima degli intervalli di confidenza al 95% mediante Block Bootstrap ($B = 10.000$ campioni). La Tabella 4.3 riassume i miglioramenti percentuali medi consolidati.

I dati della tabella confermano la robustezza del sistema su tutto l'arco operativo: il metodo registra una variazione di percorrenza del -33.72% rispetto a Nearest Neighbor e del -45.89% rispetto a LLM-Direct, migliorando contemporaneamente il tasso di puntualità tra il $+2.17\%$ e il $+11.87\%$. Tutte le differenze sono statisticamente significative ($p < 0.01$ contro i metodi euristici e generativi, e $p < 0.05$ contro il solutore esatto), provando che l'eccellenza prestazionale deriva direttamente dai fondamenti matematici dell'ipergrafo e non dalla variabilità dei campioni.

Tabella 4.3: Miglioramento relativo percentuale (Δ%) del framework a ipergrafo rispetto alle metodologie di riferimento e significatività statistica con test di Wilcoxon su benchmark multi-finestra.

Metodo di Riferimento	Δ% **Puntualità** ↑	Δ% **Accettazione** ↑	Δ% **Km/Stop** ↓	**Wilcoxon** p-value
vs *Nearest Neighbor (NN)*	+6.80%	+6.88%	-33.72%	$p < 0.01$
vs *Projection-Only*	+4.43%	+4.48%	-25.33%	$p < 0.01$
vs *Solver VRPTW (Ricerca Operativa)*	+2.17%	+2.19%	-14.50%	$p < 0.05$
vs *LLM-Direct*	+11.87%	+12.01%	-45.89%	$p < 0.01$

4.3 L'impatto dell'orchestratore agentico sui vincoli operativi testuali

Una delle innovazioni risiede nell'integrazione di un Orchestratore Agentico, capace di interpretare ed elaborare informazioni non strutturate provenienti dal linguaggio naturale.

4.3.1 La cecità dei metodi tradizionali verso le note non strutturate

Nei sistemi di consegna reali, i dati associati a una spedizione non si limitano a numeri freddi (latitudine, longitudine e finestre orarie). Nella stragrande maggioranza dei casi, i clienti o gli addetti al magazzino inseriscono istruzioni testuali in formato libero, quali ad esempio:

- «*Consegna al terzo piano senza ascensore, pacco pesante;*»
- «*Banchina di scarico merci accessibile solo entro le ore 16:00;*»
- «*Citofono guasto, chiamare al telefono all'arrivo;*»
- «*Accesso carrabile interdetto per lavori stradali, entrare a piedi dal retro.*»

I solutori classici della Ricerca Operativa (come OR-Tools) e le euristiche su grafo sono **completamente ciechi rispetto al testo**: non possiedono moduli di comprensione semantica del linguaggio naturale e sono costretti a ignorare queste note. Il solutore assegna il pacco al corriere basandosi unicamente sulle coordinate spaziali, ma quando il corriere arriva sul posto e legge la nota, si trova nell'impossibilità materiale di completare la consegna nei tempi previsti.

Nel nostro sistema, l'Orchestratore LLM riceve in ingresso lo stato corrente delle zone operative e analizza gli embedding semantici delle note operative non strutturate associate ai pacchi contesi.

Qualora l'agente rilevi una situazione critica (ad esempio, un quartiere caratterizzato da un'elevata concentrazione di consegne pesanti ai piani alti con strade strette), esso agisce modulando i parametri operativi di sensibilità o emettendo un'azione esplicita di mobilitazione dei corrieri di riserva.

L'impatto sul campo è stato quantificato misurando il tasso di veto operativo da parte dei corrieri (ovvero la percentuale di compiti assegnati che i rider sul territorio si trovano costretti a rifiutare o richiedere di riassegnare per sopravvenuta infattibilità logistica non catturata dai vincoli numerici puri):

- I metodi tradizionali e il modello privo di agente registrano un tasso fisso di veto operativo del **29.6%**, causato dalla totale cecità verso le condizioni descritte nel testo delle note.

- Con l'introduzione dell'Orchestratore Agentico, il tasso di rifiuto da parte dei corrieri scende al **22.0%** (con un abbattimento relativo del 25.7%).

Tale indicatore si distingue concettualmente dal tasso di sovrascrittura del supervisore di centrale (analizzato nel Capitolo 3 e risultato nullo grazie alla totale conformità ai guardrail): qui si misura l'adattabilità dell'assegnazione rispetto alla realtà fisica riscontrata direttamente dagli operatori sul campo.

Questo risultato dimostra come il modello linguistico consenta al sistema logistico di percepire la complessità del mondo reale che sfugge alle matrici numeriche pure, migliorando sensibilmente la fluidità operativa sul territorio.

4.4 Studio di un caso d'uso operativo

Per apprezzare in modo tangibile e intuitivo il comportamento sul campo dei diversi modelli, consideriamo uno scenario operativo tratto dalla città di Jilin durante una giornata di test.

4.4.1 Descrizione dello scenario e arrivo dell'evento dinamico

Siamo alle ore **14:30**. Nel distretto commerciale centrale di Jilin (Zona AOI 4), tre corrieri (R_1, R_2, R_3) sono già in movimento lungo i loro percorsi mattutini e hanno completato circa il 60% delle consegne previste. Improvvisamente, l'infrastruttura riceve quasi contemporaneamente cinque nuove richieste di ritiro espresso ($P_1, \ldots, P_5$), originate da due complessi direzionali contigui, con finestre di ritiro comprese tra le 14:45 e le 15:15.

4.4.2 Confronto delle risposte operative generate dai modelli

Esaminiamo come ciascuna metodologia reagisce a questo evento:

- **Cosa fa il Nearest Neighbor:** Individua il corriere fisicamente più vicino in quel preciso secondo (R_1, situato a soli 50 metri dall'edificio) e gli assegna in blocco tutti i cinque ritiri. R_1 è costretto a interrompere il suo giro, parcheggiare e gestire il carico di cinque pacchi, impiegando oltre 25 minuti. Di conseguenza, le successive sei consegne programmate da R_1 per le 15:00 falliscono tutte con ritardi.

- **Cosa fa il Projection-Only:** A causa dello schiacciamento delle relazioni su grafo 2D, l'algoritmo non riconosce che i cinque ritiri appartengono alla stessa comunità di lavoro di R_1 e R_2. Assegna tre pacchi a R_1 e due pacchi al corriere R_7, che in quel momento si trova a oltre 4.2 km di distanza in un'altra zona. R_7 è costretto a percorrere chilometri nel traffico, arrivando quando la finestra del cliente è già scaduta.

- **Cosa fa la pipeline a ipergrafo con orchestratore:**

  1. Il Dynamic Matcher interroga la matrice di incidenza dell'ipergrafo e rileva che la Zona 4 possiede un'alta densità di intersezioni.
  2. Verifica i margini orari residui (*slack*) dei corrieri attivi: R_1 ha margine per assorbire al massimo due ritiri senza violare le consegne successive, mentre R_2 può prenderne comodamente altri due.
  3. Per il quinto pacco residuo, il modello rileva che un ulteriore inserimento provocherebbe una congestione. L'Orchestratore LLM, percependo il picco di carico tramite i dati di stato, emette l'indicazione di mobilitare un corriere di riserva situato nell'hub limitrofo.

L'esito finale è che il sistema a ipergrafo completa tutti i ritiri e tutte le consegne con zero ritardi, percorrendo il tragitto minimo necessario ed evitando il sovraccarico dei lavoratori.

CAPITOLO 5

Discussione e conclusioni

In questo capitolo conclusivo vengono presentate una discussione critica e una sintesi complessiva dell'attività di ricerca svolta nella presente tesi. La prima sezione esamina nel dettaglio i risultati operativi ottenuti, i compromessi ingegneristici tra efficienza e rispetto temporale, il valore aggiunto dell'integrazione agentica, l'impatto sul benessere dei corrieri e i limiti intrinseci dell'approccio. La seconda sezione delinea le conclusioni generali, riassumendo i contributi metodologici originali del lavoro e tracciando le direttrici più promettenti per i futuri sviluppi scientifici e tecnologici.

5.1 Discussione

La progettazione e la validazione del sistema di pianificazione logistica sviluppato in questo lavoro offrono molteplici spunti di riflessione, che travalicano la pura dimensione algoritmica per toccare aspetti operativi, umani, ecologici e metodologici. La logistica dell'ultimo miglio si è rivelata un dominio di straordinaria complessità, nel quale l'efficacia di una strategia non può essere misurata unicamente in termini di pura minimizzazione geometrica dei tragitti, ma richiede una comprensione della dimensione temporale, della morfologia urbana e delle condizioni reali di lavoro sul territorio.

5.1.1 Riflessioni operative sui paradigmi a confronto

Uno degli insegnamenti più significativi emersi dall'indagine riguarda la sostanziale inadeguatezza delle strategie puramente spaziali quando applicate alla complessità cittadina. Nella pratica logistica è frequente la tentazione di affidarsi a logiche intuitive o euristiche avide basate sulla scelta della consegna fisicamente più vicina. Tuttavia, tale impostazione si è dimostrata affetta da una pericolosa miopia temporale: servire una fermata solo perché limitrofa ignora i tempi fissi necessari per ciascuna sosta (la ricerca del parcheggio, la salita ai piani, l'interazione con il cliente e la consegna vera e propria). Concentrare avidamente l'attività in una singola area ristretta per azzerare gli spostamenti immediati finisce per immobilizzare il veicolo, consumando i margini orari e innescando ritardi a catena che compromettono irrimediabilmente il rispetto delle finestre temporali per il resto della giornata.

D'altra parte, il ricorso ai solutori esatti della Ricerca Operativa classica, per quanto formalmente ineccepibile sotto il profilo dell'ottimalità teorica, si scontra con una barriera

computazionale invalicabile. La crescita combinatoria e la richiesta di tempi di calcolo nell'ordine di ore rendono tali strumenti del poco pratici nel momento in cui la giornata lavorativa è già avviata e occorre reagire tempestivamente a nuove richieste di ritiro dinamico generate dagli utenti in tempo reale.

Allo stesso modo, la semplificazione delle relazioni logistiche all'interno di grafi tradizionali a coppie si è rivelata limitante. Comprimere la complessa interazione che lega simultaneamente un corriere, una specifica zona di competenza, un deposito e un insieme eterogeneo di compiti comporta una perdita irreparabile di coesione topologica, traducendosi in rotte frammentate e percorsi disordinati che costringono i veicoli a continui e ingiustificati attraversamenti della città.

L'approccio basato su ipergrafo e articolato sui due orizzonti decisionali dimostra come sia possibile superare queste contrapposizioni. Riconoscere le relazioni ad alto ordine consente di aggregare i compiti in comunità di lavoro compatte e coerenti durante la pianificazione mattutina, mentre la delega della fase pomeridiana a una procedura locale di inserimento rapido garantisce una reattività immediata, preservando la puntualità complessiva senza sacrificare l'ordine logico e la fluidità dei tragitti.

5.1.2 Integrazione tra modelli matematici e agenti AI

I modelli generativi puri risultano inadatti all'ottimizzazione logistica autonoma, poiché privi delle strutture logico-matematiche necessarie a garantire vincoli geometrici, limiti di carico e distanze reali, incorrendo frequentemente in allucinazioni. L'approccio proposto supera tale limite separando i ruoli: un livello topologico governa rigorosamente la fattibilità fisica (distanze stradali, orari e capacità), mentre un livello neurale di supervisione interpreta segnali ad alto livello, carichi della flotta e istruzioni non strutturate. Guidato da guardrail deterministici, l'agente linguistico opera esclusivamente all'interno di un perimetro validato, unendo flessibilità semantica e garanzie formali di stabilità sul campo.

5.1.3 Impatto umano, sociale e sostenibilità urbana

L'ottimizzazione dell'ultimo miglio acquisisce valore autentico solo quando si traduce in un miglioramento concreto per le persone e per l'ambiente cittadino. I corrieri non sono semplici punti su una mappa, ma operatori soggetti a ritmi intensi e compiti usuranti. In quest'ottica, la ricerca evidenzia due benefici cruciali: da un lato, una ripartizione equilibrata del carico che previene sovraccarichi ed esaurimento psicofisico tra gli operatori; dall'altro, la capacità di interpretare le note di consegna in linguaggio naturale (come l'assenza di ascensori o limitazioni di accesso), evitando l'assegnazione di compiti irrealizzabili e riducendo frustrazioni e rifiuti sul campo.

A questi risvolti umani si unisce una diretta ricaduta ecologica: eliminando tragitti disordinati e corse a vuoto, la sensibile riduzione delle percorrenze complessive contribuisce a decongestionare il traffico cittadino e ad abbattere le emissioni inquinanti e acustiche, promuovendo una mobilità più compatibile con la vivibilità delle città moderne.

5.1.4 Contesti di massima efficacia e limiti metodologici dell'indagine

Un'analisi accademica richiede di riconoscere con obiettività sia i contesti in cui il framework esprime il proprio massimo valore applicativo, sia i limiti intrinseci dell'attuale configurazione:

1. **Ambiti di massima efficacia del modello:** L'architettura proposta eccelle in contesti metropolitani densi e complessi, caratterizzati da elevati volumi di ordini concentrati,

flotte numerose da coordinare a livello di comunità territoriali, frequenti istruzioni operative non strutturate ed eventi dinamici pomeridiani. Al contrario, in ambienti rurali o a bassissima densità abitativa, dove le consegne sono rade e distribuite su distanze molto ampie, la complessità dell'ipergrafo manca della massa critica necessaria: in tali contesti una semplice euristica geografica affiancata da un comune navigatore GPS risulta già del tutto adeguata.

2. **Stima della velocità basata su grafo statico:** Nell'attuale implementazione, i tempi di percorrenza sono derivati dalle velocità medie stradali fornite dalla cartografia OpenStreetMap tramite OSMnx. Sebbene solida ed efficace per la pianificazione generale, questa rappresentazione non incorpora ancora previsioni o flussi continui in tempo reale su code improvvise, condizioni meteorologiche avverse o chiusure temporanee per cantieri e sinistri.

3. **Omogeneità della flotta veicolare:** Il modello considera corrieri dotati delle medesime capacità di carico e della stessa velocità media di crociera. Nella logistica reale, le flotte moderne sono eterogenee e comprendono mezzi agili come biciclette cargo per i centri storici a traffico limitato, scooter per i piccoli colli e furgoni di diversa cubatura per spedizioni ingombranti.

4. **Dipendenza da connettività per la supervisione strategica:** Sebbene l'inserimento dinamico pomeridiano avvenga localmente a latenza impercettibile senza richiedere calcoli pesanti, le consultazioni dell'agente supervisore macroscopico necessitano della connessione a modelli linguistici remoti via rete. Ciò introduce una lieve latenza asincrona per i soli cicli di supervisione globale e comporta una dipendenza dalla stabilità della connessione esterna.

5.1.5 Sviluppi futuri

Le limitazioni individuate e le riflessioni operative maturate tracciano la strada per promettenti sviluppi futuri, delineando tre traiettorie di grande rilievo scientifico e applicativo:

1. **Gestione di flotte eterogenee e vincoli di transizione ecologica:** Un'estensione immediata riguarda l'adattamento della struttura dell'ipergrafo per supportare contemporaneamente flotte miste composte da scooter e furgoni elettrici. Modellando il livello di carica della batteria come uno stato dinamico dei nodi veicolo, l'ipergrafo potrà incorporare iperarchi dedicati alle colonnine di ricarica pubbliche e private, ottimizzando congiuntamente l'instradamento dei pacchi e le soste per il rifornimento energetico, nel pieno rispetto dell'autonomia residua dei mezzi a emissioni zero.

2. **Sistemi multi-deposito e stima dinamica del traffico con Digital Twin:** Una seconda direttrice prevede l'integrazione di flussi di traffico in tempo reale e previsioni meteorologiche, aggiornando dinamicamente i pesi della rete stradale tramite un vero e proprio gemello digitale (*Digital Twin*) della città. Inoltre, il modello può essere esteso alla gestione simultanea e coordinata di depositi multipli periferici e micro-hub urbani distribuiti capillarmente nei quartieri, abilitando operazioni di trasbordo intermedio delle merci per ridurre ulteriormente le distanze percorse dai veicoli pesanti nei centri storici.

3. **Modelli linguistici multimodali ed Edge AI sui dispositivi palmari:** Sul fronte dell'Intelligenza Artificiale, una prospettiva di forte impatto riguarda l'evoluzione dell'Orchestratore verso modelli multimodali, capaci di elaborare congiuntamente testo e immagini, e l'integrazione di modelli linguistici compatti eseguiti direttamente sui

terminali palmari o sugli smartphone in dotazione ai corrieri. Il corriere potrebbe trasmettere la fotografia di un citofono illeggibile, di un cartello con orari speciali o di una banchina di carico; l'agente interpreterebbe visivamente il contesto, suggerendo in tempo reale il punto di sosta ottimale e aggiornando istantaneamente il grafo operativo. L'esecuzione locale a bordo del dispositivo azzererebbe la latenza di trasmissione, eliminerebbe i costi di inferenza via rete, tutelerebbe la riservatezza dei dati e garantirebbe la piena continuità del servizio anche in gallerie, parcheggi sotterranei o aree urbane prive di connettività cellulare.

5.2 Conclusioni

Nel corso di questa tesi abbiamo affrontato la complessa sfida della logistica dell'ultimo miglio, esplorando la convergenza tra due paradigmi tradizionalmente distinti ma straordinariamente complementari: la matematica discreta applicata all'analisi delle reti complesse ad alto ordine e l'Intelligenza Artificiale basata su modelli linguistici di grandi dimensioni. L'intero percorso è stato guidato dalla volontà di superare il duplice ostacolo che frena la disciplina: da un lato l'esplosione combinatoria e la rigidità relazionale dei solutori della Ricerca Operativa classica, dall'altro la fragilità e le allucinazioni dei modelli generativi puri, del tutto inaffidabili se lasciati privi di vincoli fisici.

Per rispondere a tale criticità, abbiamo formalizzato un'architettura integrata fondata su un ipergrafo tipizzato e pesato, articolata su due orizzonti decisionali cooperanti. Nella fase iniziale di pianificazione mattutina abbiamo modellato le complesse relazioni multi-entità tra corrieri, aree di interesse, depositi e compiti di consegna senza compressioni topologiche, impiegando processi di diffusione markoviana per identificare l'affinità territoriale e risolvendo la formazione di comunità stabili mediante l'algoritmo di Leiden e tecniche di ottimizzazione locale.

Parallelamente, per far fronte all'arrivo imprevedibile delle richieste pomeridiane in streaming, abbiamo sviluppato un motore di inserimento, capace di operare a livello locale in tempi pressoché istantanei senza richiedere il ricalcolo globale delle rotte. A presidio dell'intero sistema abbiamo introdotto un Orchestratore Agentico basato su modelli linguistici, demandandogli la supervisione strategica della flotta e la comprensione semantica delle istruzioni espresse in testo libero dai clienti. L'affidabilità industriale del modello è stata garantita vincolando rigorosamente ogni sua azione all'interno di una sequenza di guardrail con limiti geometrici di sicurezza, escludendo matematicamente ogni rischio di allucinazione operativa o violazione dei vincoli territoriali.

La successiva fase di sperimentazione su dati reali tratti dal dataset industriale Cainiao LaDe ha confermato la validità e l'equilibrio della soluzione. Il sistema ha dimostrato di saper garantire un'elevata puntualità complessiva, riducendo drasticamente le percorrenze chilometriche rispetto alle proiezioni tradizionali, distribuendo in modo equo e sostenibile il carico di lavoro tra tutti gli operatori e abbattendo sensibilmente il tasso di rifiuto delle consegne sul campo grazie all'interpretazione tempestiva delle note informative non strutturate.

La trasformazione digitale della mobilità urbana e della logistica distributiva rappresenta ormai un passaggio ineludibile per la sostenibilità economica ed ecologica dei prossimi decenni. Il cammino finora percorso dalla ricerca scientifica è stato spesso caratterizzato dalla contrapposizione tra la certezza dei modelli analitici chiusi e la flessibilità dell'apprendimento automatico. Tuttavia, l'evoluzione verso sistemi decisionali realmente capaci di operare nel mondo reale richiede una sintesi più matura e cooperativa. Come evidenziato in questo lavoro, porre il rigore delle strutture matematiche a presidio della fattibilità fisica ed elevare l'Intelligenza Artificiale al ruolo di interprete strategico del contesto umano consente di coniugare l'efficienza operativa con la salvaguardia dell'ambiente e il rispetto dei lavoratori.

In conclusione, il futuro dei trasporti intelligenti non risiede nella sostituzione dell'uomo o in una cieca automazione algoritmica, bensì nello sviluppo di strumenti trasparenti capaci di valorizzare, proteggere e supportare l'operato umano sul territorio.

Bibliografia

ARUNKUMAR, V., GANGADHARAN, G. R. e BUYYA, R. (2026), «Agentic Artificial Intelligence (AI): Architectures, Taxonomies, and Evaluation of Large Language Model Agents», *arXiv preprint arXiv:2601.12560*. (Cited at page 10)

BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R. e LEFEBVRE, E. (2008), «Fast unfolding of communities in large networks», *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008 (10), p. P10 008. (Cited at page 7)

BOEING, G. (2017), «OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks», *Computers, Environment and Urban Systems*, vol. 65, p. 126–139. (Cited at page 26)

BRETTO, A. (2013), *Hypergraph Theory: An Introduction*, Mathematical Engineering, Springer International Publishing. (Cited at pages 5 e 15)

FROHNER, N. e RAIDL, G. R. (2020), «A Double-Horizon Approach to a Purely Dynamic and Stochastic Vehicle Routing Problem with Delivery Deadlines and Shift Flexibility», *Algorithms and Complexity Group, TU Wien*. (Cited at page 4)

KERNIGHAN, B. W. e LIN, S. (1970), «An efficient heuristic procedure for partitioning graphs», *The Bell System Technical Journal*, vol. 49 (2), p. 291–307. (Cited at pages 8 e 23)

KIVELÄ, M., ARENAS, A., BARTHELEMY, M., GLEESON, J. P., MORENO, Y. e PORTER, M. A. (2014), «Multilayer networks», *Journal of Complex Networks*, vol. 2 (3), p. 203–271. (Cited at pages 5 e 14)

PAGE, L., BRIN, S., MOTWANI, R. e WINOGRAD, T. (1999), «The PageRank Citation Ranking: Bringing Order to the Web», Technical Report 1999-66, Stanford InfoLab. (Cited at page 8)

SCHULMAN, J., LEVINE, S., MORITZ, P., JORDAN, M. I. e ABBEEL, P. (2015), «Trust Region Policy Optimization», in «International Conference on Machine Learning (ICML)», p. 1889–1897. (Cited at pages 11 e 25)

SINGH, S. S., SRIVASTAVA, D., VERMA, M. e MUHURI, S. (2024), «Information diffusion analysis: process, model, deployment, and application», *The Knowledge Engineering Review*, vol. 39, p. e11. (Cited at page 8)

SOLOMON, M. M. (1987), «Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints», *Operations Research*, vol. 35 (2), p. 254–265. (Cited at pages 3 e 39)

TOTH, P. e VIGO, D. (2014), *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), 2nd ed. (Cited at pages 3 e 38)

TRAAG, V. A., WALTMAN, L. e VAN ECK, N. J. (2019), «From Louvain to Leiden: guaranteeing well-connected communities», *Scientific Reports*, vol. 9 (1), p. 5233. (Cited at pages 7 e 21)

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł. e POLOSUKHIN, I. (2017), «Attention Is All You Need», in «Advances in Neural Information Processing Systems (NeurIPS)», vol. 30, p. 5998–6008. (Cited at page 9)

WEI, J., WANG, X., SCHUURMANS, D., BOSMA, M., ICHTER, B., XIA, F., CHI, E., LE, Q. e ZHOU, D. (2022), «Chain-of-Thought Prompting Elicits Reasoning in Large Language Models», in «Advances in Neural Information Processing Systems (NeurIPS)», vol. 35, p. 24 824–24 837. (Cited at page 9)

WILCOXON, F. (1945), «Individual Comparisons by Ranking Methods», *Biometrics Bulletin*, vol. 1 (6), p. 80–83. (Cited at pages 29 e 42)

YAO, S., ZHAO, J., YU, D., DU, N., SHAFRAN, I., NARASIMHAN, K. e CAO, Y. (2023), «ReAct: Synergizing Reasoning and Acting in Language Models», in «International Conference on Learning Representations (ICLR)», . (Cited at page 10)

ZHANG, L., WANG, S., LIU, Y., ZHANG, P. e DU, B. (2024), «LaDe: The First Comprehensive Last-mile Delivery Dataset from Industry», in «Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks», vol. 37. (Cited at pages 26 e 40)

Siti web e repository consultati

- Google OR-Tools Routing Library – `https://developers.google.com/optimization/routing`
- OpenStreetMap & OSMnx Project – `https://osmnx.readthedocs.io`
- Cainiao LaDe Open Dataset – `https://github.com/cainiao-ai/LaDe`