



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Minimizzazione del rumore nella scelta delle traiettorie di decollo

TESI DI LAUREA MAGISTRALE IN
COMPUTER SCIENCE AND ENGINEERING
INGEGNERIA INFORMATICA

Author: **Luigi V De Fatico**

Student ID: 921363

Advisor: Prof. Federico Malucelli

Co-advisor: Ing. Matteo Crippa

Academic Year: 2020-21

Abstract

An airport nearby a residential area always means noise for the neighboring population. Over the years, the authorities have studied various methods to measure and reduce the annoyance perceived by the inhabitants of these areas.

This thesis work aims to offer a tool that can help to mitigate this problem through an innovative approach based on a generative algorithm that can produce a trajectory that minimize the noise perceived by the population.

The work will focus firstly on the state-of-the-art analysis in order to identify problems and possibilities of development.

Some changes will be proposed based on the analysis results, in order to expand the adopted model and to improve the implementation performances, aiming to increase the generated trajectory effectiveness.

The changes will be implemented to allow their validation through a process of testing.

Finally, two case studies based on Malpensa airport will be presented, in which the performance of the algorithm will be compared before and after the changes' introduction.

Key-words: noise, airport, aircraft, optimization, generative algorithm, A*, Dijkstra's algorithm

Abstract in italiano

La presenza di un aeroporto in un'area abitata è da sempre sinonimo di rumore per la popolazione limitrofa.

Negli anni, le autorità hanno studiato diversi metodi per misurare e ridurre il fastidio percepito dagli abitanti di tali aree.

L'obiettivo di questa tesi è offrire uno strumento atto a mitigare il problema tramite un innovativo approccio basato su un algoritmo generativo finalizzato a produrre una traiettoria che minimizzi il rumore percepito dalla popolazione.

Lo studio si concentrerà, in una prima fase, sull'analisi dello stato dell'arte al fine di individuarne problematiche e possibilità di sviluppo.

Sulla base di tale analisi, verranno proposte delle modifiche atte ad espandere il modello utilizzato e migliorare le prestazioni dell'implementazione, con lo scopo di rendere più efficaci le traiettorie generate.

Le modifiche verranno implementate al fine di renderne possibile la validazione dei miglioramenti ottenuti.

Infine, verranno presentati due casi studio basati sull'aeroporto di Malpensa in cui verranno confrontate le prestazioni dell'algoritmo prima e dopo l'introduzione delle modifiche.

Parole chiave: rumore, aeroporto, velivolo, ottimizzazione, algoritmo generativo, A*, algoritmo di Dijkstra

Indice

Abstract	iii
Abstract in italiano	v
Indice	vii
1 Introduzione	1
2 Il Rumore Aeronautico	7
2.1. Inquinamento Acustico.....	7
2.2. La misura del suono	8
2.2.1. Metriche per il rumore.....	9
3 Il problema	11
3.1. Approccio Bilanciato	11
3.1.1. Riduzione del rumore alla fonte	12
3.1.2. Pianificazione d'uso del terreno circostante.....	12
3.1.3. Procedure operative di abbattimento del rumore	12
3.1.4. Restrizioni operative.....	13
3.1.5. Coinvolgimento delle parti interessate	13
3.2. La soluzione proposta.....	14
4 Base di partenza	15
4.1. Stato dell'Arte.....	15
4.1.1. Modellazione del contesto	15
4.1.2. Generazione della traiettoria	16
4.1.3. Ottimizzazione della rotta.....	18
4.1.4. Considerazioni.....	19
4.2. Analisi dello stato attuale	21
4.2.1. Velocità dell'algoritmo generativo	21
4.2.2. Comparazione delle prestazioni tra A* e algoritmo di Dijkstra...	22

4.2.3.	Problema sulla giunzione tra i due alberi nella ricerca bidirezionale	28
4.2.4.	Problemi minori.....	35
5	Il nuovo approccio	39
5.1.	Distanza minima tra due nodi	39
5.1.1.	Percorsi di Dubins	39
5.1.2.	Implementazione dei percorsi.....	41
5.1.3.	Applicazione dei percorsi di Dubins.....	41
5.2.	Introduzione terza dimensione	43
5.2.1.	Crescita costante	45
5.2.2.	Problemi implementativi.....	45
5.3.	Taglio rumore con passa-alto.....	48
5.3.1.	Modello di rumore attuale	49
5.3.2.	Modello di rumore direzionale	53
5.3.3.	Sinergia con il calcolo dell'euristica.....	54
5.4.	Modifica dell'euristica.....	55
5.4.1.	Aumento della distanza minima tramite percorsi di Dubins	55
5.4.2.	Cambio di modello.....	57
5.5.	Definizione dinamica dell'area di ricerca	61
5.5.1.	Descrizione del processo	62
6	Risultati computazionali.....	63
6.1.	Aeroporto di Malpensa.....	63
6.1.1.	Le piste	63
6.1.2.	Procedure di abbattimento del rumore attive.....	63
6.2.	Hardware e software utilizzato	65
6.3.	Casi studio	66
6.3.1.	Partenza SID DOGUB5L	66
6.3.2.	Partenza SID SRN5W.....	72
7	Conclusioni.....	79
7.1.	Risultato delle analisi	79

7.1.1.	Modifiche al modello	79
7.1.2.	Aumento prestazionale	80
7.1.3.	Considerazioni generali.....	80
7.2.	Sviluppi futuri.....	82
7.2.1.	Ottimizzazioni computazionali.....	82
7.2.2.	Espansione modello	83
Bibliografia	85
Lista delle Figure	87
Lista delle Tabelle	89

1 Introduzione

La presenza di un aeroporto è da sempre associata alla percezione di continui rumori da parte degli abitanti delle zone limitrofe e, per questo motivo, non gradita dagli stessi.

Ciò è causato dal transito dei velivoli che continuamente atterrano o decollano dal sito, attraversando lo spazio aereo circostante ad un'altitudine necessariamente molto minore comparata alla quota di crociera, massimizzandone l'impatto acustico.

L'esposizione continua al rumore genera nell'uomo effetti sulla salute che spaziano dalla percezione di un senso di fastidio alla comparizione di disturbi del sonno, fino ad effetti più severi come il malfunzionamento di organi e apparati.

A causa di ciò, le autorità hanno provato, negli anni, a misurare e contrastare la comparizione di tale fenomeno. L'approccio attualmente utilizzato prevede di ridurre il problema affrontandolo da diverse prospettive, generando un compromesso tra tutti i diretti interessati, i quali presentano problematiche ed interessi differenti tra loro, al fine sia di ottenere un risultato migliore sia di non gravare eccessivamente su nessuno di essi. Tale modalità prende il nome di Approccio Bilanciato e rappresenta a livello internazionale lo standard odierno per quanto riguarda l'abbattimento del rumore nelle fasi di decollo e atterraggio.

Esso mira ad affrontare il problema da quattro fronti contemporaneamente. Il primo è la riduzione del rumore alla fonte, che mira a sfruttare gli avanzamenti tecnologici per ridurre sempre di più il rumore generato dai velivoli. Quello dell'inquinamento acustico è, infatti, un problema principale per l'industria aeronautica, che punta a progettare motori sempre più silenziosi e dotare i mezzi di sistemi di isolamento acustico all'avanguardia che permettono al contempo di non aumentare eccessivamente il peso complessivo del velivolo.

Il secondo punta ad imporre un uso specifico per i terreni circostanti che permetta di limitare l'impatto acustico sulle persone.

Vengono utilizzate, infine, delle procedure nelle fasi di decollo e atterraggio che limitano il rumore e limitate, nei casi peggiori, le possibilità operative di alcuni aeroporti più problematici.

Il lavoro presentato in questa tesi è finalizzato ad offrire uno strumento matematico che permetta di ottimizzare le procedure seguite nella fase di decollo e atterraggio. In particolare, generando una rotta che permetta ai velivoli di far percepire il minor rumore possibile sui centri abitati vicini ad un aeroporto, minimizzandone l'impatto totale.

La base di partenza è rappresentata da uno degli unici algoritmi incentrati sull'ottimizzazione delle rotte aeree orientata a questo scopo.

Esso utilizza due approcci distinti per generare un percorso ottimizzato, il primo in grado di generare una rotta e il secondo di modificarne una esistente in modo da ridurre il rumore totale.

L'algoritmo, però, seppur promettente, offre attualmente delle prestazioni che non lo rendono compatibile con l'uso in un contesto realistico.

Esso funziona con un modello molto semplificato del volo di un velivolo. Quest'ultimo è considerato come se fosse posto sempre all'altezza del suolo, il che ne pregiudica la capacità di generazione del percorso ottimale essendo diverse, ad altezze differenti, le rotte che un aereo dovrebbe seguire per minimizzare il rumore totale.

In più, il modo in cui il rumore viene calcolato considera una funzione matematica approssimativa che tiene conto solo della distanza dai *recettori sensibili*, il che ne pregiudica ulteriormente la correttezza della soluzione.

Inoltre, il tempo impiegato per ottenere un risultato è eccessivo a causa di prestazioni poco soddisfacenti e molto dipendenti dal contesto geografico.

Lo scopo di questa tesi è di creare, partendo da queste basi, un algoritmo in grado di generare con maggiore precisione una soluzione corretta in un contesto geografico qualsiasi, utilizzando un modello più realistico e migliorando le prestazioni generali dello stesso.

La nuova versione integra l'altitudine nello stato del velivolo, offrendo una soluzione basata sull'incremento costante della stessa, soluzione che è al contempo in grado di trovarsi in linea con il contesto di decollo (o atterraggio, al contrario) e di non aumentare il carico computazionale necessario a trovare una soluzione.

È stata, inoltre, resa più realistica la funzione di rumore, integrando nel calcolo la direzione del velivolo oltre che la sua distanza dai recettori sensibili.

Sono state, inoltre, proposte diverse modifiche al fine di rendere più basso il numero di iterazioni dell'algoritmo necessarie per trovare una soluzione e il tempo impiegato.

Prestazioni non ottimali portano ad aumentare i compromessi necessari per arrivare a calcolare una soluzione.

L'obiettivo di migliorare prestazionalmente l'algoritmo è quindi anche collegato alla limitazione dell'impatto delle approssimazioni presenti sulla correttezza della traiettoria, diminuendo i compromessi necessari ad arrivare a terminazione in un tempo accettabile.

In primo luogo, sono stati analizzati i componenti dell'algoritmo per valutarne l'efficienza nel contesto studiato.

Un esempio è dato dallo studio sull'impatto prestazionale dell'euristica e della ricerca bidirezionale.

Successivamente si è cercato di limitare l'impatto di ogni iterazione dell'algoritmo sul tempo finale, riducendo la quantità di calcoli necessari a completarla. Il tempo per ogni iterazione aumenta di pari passo con l'accrescere del numero di elementi totali considerati nella sua esecuzione, come il numero di recettori sensibili.

I risultati ottenuti in questa tesi hanno portato ad un netto miglioramento dell'algoritmo, che è ora in grado di generare traiettorie ottimizzate su un modello più realistico e quindi più prossime ad un'ottimizzazione del caso reale. Inoltre, l'algoritmo offre ora una maggior indipendenza dal contesto di partenza, offrendo un maggior livello di generalizzazione in contesti in cui precedentemente, a causa della disposizione dei recettori sensibili, era necessario intervenire manualmente imponendo ulteriori limitazioni.

I vantaggi prestazionali offerti dalle introduzioni proposte sono stati utilizzati per spingere più in là la correttezza delle traiettorie prodotte nella fase di generazione delle stesse, su cui questo lavoro si è concentrato.

In particolare, l'analisi ha toccato sia aspetti che coinvolgono la modellizzazione del contesto, sia considerazioni che hanno portato ad un miglioramento prestazionale. Della prima categoria fanno parte l'analisi delle modalità di movimento di velivolo che hanno portato alla ridefinizione del concetto di distanza minima, che ha trovato diverse applicazioni nella tesi.

Inoltre, è stato introdotto il modello tridimensionale di spostamento, fondamentale per la correttezza delle traiettorie generate.

La ricerca di efficienza ha portato a riconsiderare l'apporto dell'euristica in tal senso, fornendo un approccio che ne riconsidera le modalità di calcolo e il suo valore. A tal scopo è stato analizzato ulteriormente l'impatto che la quantità di recettori ha sulle prestazioni arrivando a definire una modalità di calcolo che consideri esclusivamente quelli rilevanti in modo da ridurre l'effetto negativo sui tempi da essi scaturito.

L'algoritmo è stato codificato in Python ed è utilizzabile per generare una traiettoria minima utilizzando le varie modalità definite in questa tesi a cui l'utente può accedere tramite un file di configurazione.

Il programma prende in ingresso una configurazione iniziale che modella il territorio circostante e genera una traiettoria minima formata da punti intermedi.

Nei prossimi capitoli è presentato lo studio effettuato per raggiungere lo stato implementativo attuale.

Questa parte parte dalla definizione in maniera più approfondita del concetto di rumore aeronautico e di come le autorità hanno provato a misurarlo e gestirlo, introducendo il concetto di fastidio.

Verrà poi presentato il funzionamento, in maniera maggiormente dettagliata, dell'Approccio Bilanciato, al fine di rendere chiaro dove questo lavoro si collochi nello sforzo internazionale orientato all'abbattimento del rumore aeronautico.

Il [Capitolo 4](#) offrirà una visione più dettagliata dei principi di funzionamento dell'algoritmo, sia nella fase generativa che di ottimizzazione, spiegando i motivi dietro alle scelte implementative ed evidenziandone alcune delle problematiche ad esse legate.

L'analisi si concentra su aspetti implementativi che generano incongruenze tra i percorsi trovati al variare della configurazione iniziale.

Inoltre, verranno analizzate le prestazioni dal punto di vista dei tempi e in termini di iterazioni necessarie a trovare una soluzione, utilizzando tale analisi per giustificare le modifiche proposte successivamente.

Nel [Capitolo 5](#), quindi, sono presentate le modifiche proposte che hanno lo scopo di rendere la soluzione trovata dall'algoritmo maggiormente compatibile con il caso reale.

Viene analizzato come la loro implementazione è resa possibile da considerazioni effettuate che ne permettono l'introduzione evitando che esse modifichino in modo insostenibile le prestazioni dell'algoritmo.

Successivamente, nel [Capitolo 6](#), vengono presentati due casi di studio basati sull'aeroporto di Malpensa, il quale viene preso come riferimento nel lavoro svolto. I risultati computazionali sono utilizzati per evidenziare come i percorsi trovati dalla nuova implementazione varino in comparazione a quelle generate dall'algoritmo precedente e alle rotte ufficiali.

Lo scopo è quello di validare le modifiche inserite utilizzando un caso reale, al fine di considerare il problema da un punto di vista olistico.

Infine, nel [Capitolo 7](#) sono presentate le conclusioni a cui il lavoro svolto giunge, evidenziando gli aspetti positivi e negativi della nuova implementazione e fornendo delle idee per futuri avanzamenti dell'algoritmo.

2 Il Rumore Aeronautico

2.1. Inquinamento Acustico

L'Inquinamento Acustico è un fenomeno caratterizzato dall'eccessiva esposizione di un dato ambiente a rumori ad alta intensità.

In Italia la legge n. 447/1995 ne dà una definizione:

“L'introduzione di rumore nell'ambiente abitativo o nell'ambiente esterno tale da provocare fastidio o disturbo al riposo ed alle attività umane, pericolo per la salute umana, deterioramento degli ecosistemi, dei beni materiali, dei monumenti, dell'ambiente abitativo o dell'ambiente esterno o tale da interferire con le legittime fruizioni degli ambienti stessi”

L'esposizione prolungata a fonti di rumore può causare diversi effetti sulle persone sul lungo andare. Essa può essere fonte di danni psicologici, stress o provocare persino danni fisici. Oltre ad essere causa di una possibile perdita, parziale o totale, dell'udito, l'inquinamento acustico è collegato ad una serie di condizioni mediche che includono disordini cardiovascolari, ipertensione, incremento del livello di stress, acufene, disturbo del sonno e altri effetti dannosi. (Münzel T, 2018) (Rosen, 1965) (BBC News, 2009) (Kerns, 2018).

Un aeroporto nelle immediate vicinanze figura tra le principali cause alle spalle dell'inquinamento acustico nei moderni spazi urbani. Le aree residenziali circostanti sono caratterizzate da un frequente e prolungato rumore generato dai motori dei velivoli che transitano attraverso tali aeroporti. È da notare come gli aerei si trovino frequentemente, in tali aree, nelle fasi di atterraggio e decollo il che ne riduce l'altitudine e ne massimizza il rumore percepito a livello del suolo.

Ciò diminuisce sensibilmente la qualità della vita di chi vive in città situate nei pressi di aeroporti, nelle quali è frequentemente registrato un senso di fastidio collegato a questo fenomeno.

2.2. La misura del suono

Il decibel è l'unità utilizzata per misurare l'intensità di un suono. Esso corrisponde al modo in cui l'orecchio umano interpreta la pressione sonora. L'orecchio umano riesce a percepire pressioni sonore in un raggio molto ampio ed è, per tanto, misurato in una scala logaritmica.



Figura 2.1: Scala di riferimento decibel

L'orecchio umano tende a rispondere in modo differente a diverse frequenze sonore, riuscendo a percepire molto più chiaramente suoni caratterizzati da alte frequenze.

Per tenere conto delle differenze nel modo in cui le persone rispondono al suono, viene utilizzata una scala che da un peso diverso al livello sonoro in base alla sua frequenza (dBA).

Tale scala approssima in modo più corretto la rumorosità di un suono nell'aria percepito dall'orecchio umano e, perciò, fornisce uno strumento più efficace per valutare l'effetto dell'esposizione al rumore sugli umani, concentrandosi su quello spettro di frequenze che risulta più udibile.

Essa è lo standard *de facto* per tutte le valutazioni di livelli sonori che hanno come scopo quello di misurare il disturbo di un suono sull'uomo. (FAA, 2022)

2.2.1. Metriche per il rumore

Il Livello di Esposizione al Suono (SEL) è una metrica che rappresenta l'energia acustica, o pressione sonora, di un singolo evento sonoro come se l'intero evento avesse luogo nell'arco di un secondo.

La scala SEL cattura sia la magnitudine che la durata di un evento sonoro in una singola quantificazione numerica, inserendo tutta l'energia sonora generata da un evento in un secondo.

Tale scala è usata per riuscire a comparare in modo uniforme eventi sonori di durate differenti.

In aeronautica, la scala DNL (Day-Night average sound Level) è utilizzata al fine di riflettere il carattere cumulativo dell'esposizione al suono di un individuo su un periodo di 24 ore.

La scala esprime il livello di rumore in un giorno medio sulla base dei velivoli che transitano in un anno nell'area che circonda un aeroporto.

Nella misurazione, una penalità di 10 dBA viene applicata ai velivoli che transitano nell'area in orari compresi tra le 10pm e le 7am, che equivale a misurare il suo apporto con un peso dieci volte maggiore a quanto usato durante le ore diurne per lo stesso evento.

La scala rappresenta lo standard usato dalla Federal Aviation Administration, la quale ha adottato DNL 65 dBA come soglia ad indicare una significativa esposizione al rumore, sotto la quale è compatibile un uso dell'area a scopo residenziale (FAA, 2022).

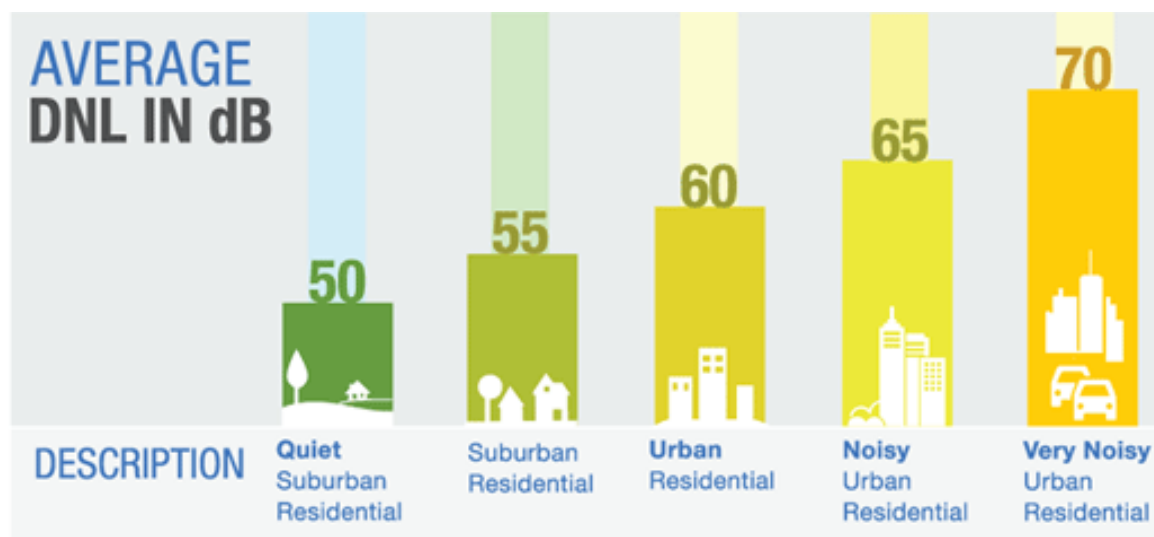


Figura 2.2: Scala di riferimento per valori DNL, espressa in dB

La Figura 2.2 sintetizza i dati risultanti da un questionario portato avanti dalla stessa FAA su un campione di 10000 abitanti di zone esposte a livelli DNL compresi tra 50 e 70.

La scala cerca, per cui, di trasporre la misura fisica del rumore al senso di fastidio percepito dalle comunità situate nelle località prossime ad aeroporti.

3 Il problema

L'incremento del traffico aereo nelle ultime decadi ha portato ad una legislazione via via più strutturata da parte delle autorità competenti.

L'ambiente normativo si è evoluto con l'introduzione di regole e procedure che hanno introdotto restrizioni operazionali legate al rumore regolamentate da un Approccio Bilanciato (EU, 2014).

Questa norma promuove delle tecniche di gestione al fine di governare l'inquinamento acustico nei pressi degli aeroporti, in maniera complementare all'implementazione di iniziative nazionali o locali.

3.1. Approccio Bilanciato

Il principio alla base di un "Approccio Bilanciato" nei confronti della gestione del rumore generato dai velivoli nei pressi di un aeroporto include la modellazione e il monitoraggio della situazione attuale, la creazione di una linea di base tramite la quale effettuare comparazioni e la definizione di obiettivi futuri associati ad un piano di azione.



Figura 3.1: punti chiave dell'Approccio Bilanciato

Il piano di azione è composto da quattro componenti di base, riassunti nella **Figura 3.1**: punti chiave dell'Approccio Bilanciato: Riduzione del rumore alla fonte, pianificazione d'uso del terreno circostante, procedure operative di abbattimento del rumore e restrizioni operative.

3.1.1. Riduzione del rumore alla fonte

Il primo punto si pone l'obiettivo di ridurre il rumore percepito agendo direttamente sulla fonte, tramite ricerca e innovazione tecnologica.

L'inquinamento acustico è infatti diventato un problema centrale per l'industria aerea, nella quale i produttori provano a costruire motori che emettono sempre meno rumore e isolandoli in maniera sempre più efficace.

La sfida maggiore è rappresentata, però, dalla necessità di bilanciare l'aumento del grado di isolamento acustico con la riduzione del peso del velivolo, che grava direttamente sul consumo di carburante.

Il fine è quello di non accentuare aspetti altrettanto importanti come l'aumento dell'inquinamento prodotto e dei costi sostenuti.

3.1.2. Pianificazione d'uso del terreno circostante

Un aspetto fondamentale è la gestione preventiva dell'area circostante all'aeroporto al fine di prevenire sviluppi incompatibili delle zone residenziali con le metriche e gli obiettivi definiti dal piano di riduzione del rumore.

È per questo che l'approccio bilanciato prevede di gestire tali zone con diverse azioni che possono prevedere:

- Progettazione urbanistica, tramite la zonizzazione dell'area;
- Mitigazione edilizia, tramite norme apposite, coibentazioni degli edifici interessati e un aumento della consapevolezza negli acquirenti;
- Aspetti finanziari, tramite incentivi e oneri fiscali legati al problema.

3.1.3. Procedure operative di abbattimento del rumore

Questo aspetto prevede la modifica delle procedure di volo dal punto di vista operativo.

Ricadono in quest'area la modifica delle traiettorie, l'implementazione di punti di uscita per quest'ultime, l'adozione di una data velocità variometrica minima e tutte le operazioni procedurali atte a ridurre o redistribuire il rumore nelle aree limitrofe al sito di decollo che fanno leva su tutte le capacità dei velivoli moderni, senza per questo sacrificare la sicurezza del volo.

3.1.4. Restrizioni operative

Fanno riferimento a questa categoria tutte le restrizioni relative all'abbattimento del rumore che limitano l'accesso o riducono la capacità operativa di un aeroporto.

A tal proposito possono essere impiegate, ad esempio, quote massime di rumore o restrizioni al numero di voli in transito per l'aeroporto.

Queste restrizioni sono solitamente messe in pratica solo in seguito alla presa in considerazione degli altri elementi dell'approccio bilanciato.

3.1.5. Coinvolgimento delle parti interessate

Il coinvolgimento di tutte le parti interessate nella discussione sull'apporto bilanciato alla gestione del rumore è un fattore critico nella mitigazione dell'inquinamento acustico e del fastidio percepito dalle comunità limitrofe.

A tal proposito, la norma europea vigente (EU, 2014) prevede che le misure di mitigazione del rumore vengano stabilite tramite una collaborazione tecnica tra gli operatori aeroportuali, i produttori di velivoli, gli operatori di terra e i fornitori di servizi di navigazione aerea.

In aggiunta, è necessario consultare i rappresentanti locali dei residenti e le autorità locali, fornendo le informazioni necessarie sulle misure di attenuazione del rumore. Questa fase è fondamentale al fine di adottare delle misure che si adattino alle differenti necessità e possibilità delle zone prese in esame.

Facilita, inoltre, l'individuazione di possibili sinergie ed è fondamentale per quantificare l'impatto che tali restrizioni operative avrebbero su altri ambiti.

3.2. La soluzione proposta

La soluzione proposta in questo lavoro trova spazio nell'approccio sopra descritto all'interno procedure operative di abbattimento del rumore.

Essa mira a ridefinire la traiettoria che un aereo dovrebbero seguire al fine di ridistribuire al meglio il rumore prodotto dai motori.

La redistribuzione avviene tramite la minimizzazione del rumore totale, inteso come integrale del rumore percepito sulle zone abitative limitrofe sulla traiettoria seguita.

3.2.1.1. Compromesso tra riduzione del rumore e consumo di carburante

L'incremento del consumo di carburante rappresenta il principale limite da tenere in considerazione nell'adozione di una traiettoria che minimizzi il rumore al suolo. Le scelte operative adottate a tale scopo impongono, infatti, una gestione delle traiettorie che si discosta dall'ottimizzazione del consumo di carburante: al fine di tenersi il più lontano possibile dalle zone residenziali è spesso necessario, come facilmente intuibile, incrementare la distanza percorsa. Tale incremento incide sul consumo di carburante.

Negli sviluppi futuri presentati in questo lavoro, tale compromesso sarà preso in considerazione al fine di non generare un percorso operazionalmente non percorribile.

4 Base di partenza

Il lavoro presentato in questo elaborato è il risultato dell'analisi di "Air Trajectory Optimization for Noise Reduction" (Gullo, 2021). Esso rappresenta uno degli unici lavori mirato a ridurre l'inquinamento acustico attraverso l'ottimizzazione delle rotte aeree.

4.1. Stato dell'Arte

4.1.1. Modellazione del contesto

L'esito del lavoro è un algoritmo che prende in ingresso due punti di passaggio e un insieme di recettori sensibili, generando una rotta che minimizza il rumore percepito dai recettori sensibili.

I punti di passaggio rappresentano il punto iniziale e finale della rotta generata. Sono caratterizzati da coppie di coordinate e una direzione che rappresentano rispettivamente la posizione geografica e l'orientamento del velivolo. Il punto iniziale rappresenta la pista di decollo di un aeroporto, mentre l'obiettivo rappresenta il punto di uscita della rotta, cioè il punto in cui il velivolo inizia a seguire un percorso precedentemente definito.

I recettori sensibili sono la rappresentazione dello strumento usato per misurare il rumore percepito in un punto specifico e sono caratterizzati a loro volta da una coppia di coordinate e da un peso.

Le coordinate, anche in questo caso, rappresentano la posizione geografica mentre il peso è uno scalare che indica la gravità dell'emissione sonora in quello specifico punto.

In questo caso, la gravità cresce linearmente con la densità abitativa dell'area che circonda il recettore sensibile.

L'obiettivo dell'algoritmo è la minimizzazione del rumore percepito sul gruppo di recettori sensibili.

Allo stato attuale, l'intensità del rumore è calcolata come se il velivolo fosse una fonte di rumore puntiforme omnidirezionale.

Per cui, per ogni recettore sensibile, il valore del rumore è calcolato come nella (4.1),

dove P è la potenza del motore, w_i è il peso del recettore e r è la distanza tra il recettore e il velivolo.

$$I = \frac{P * w_i}{4 \pi r^2} \quad (4.1)$$

L'algoritmo considera l'integrale dell'emissione sonora totale sul percorso come funzione di costo da minimizzare.

Un'approssimazione è stata inserita per ridurre la complessità computazionale dell'algoritmo, discretizzando tale integrale in una sommatoria ottenuta campionando il rumore sul percorso ad intervalli regolari della traiettoria.

Opzionalmente, è possibile specificare un set di No Flight Zone. Una NFZ è un'area in cui il velivolo non può transitare. Non è quindi possibile generare un segmento appartenente alla traiettoria che passi al suo interno.

Una traiettoria è descritta come una relazione sequenziale di stati. Questi ultimi rappresentano la posizione fisica del velivolo e sono descritti da una coppia di coordinate bidimensionali e dalla direzione del velivolo.

L'ottimizzazione può essere effettuata usando due approcci mutualmente non esclusivi: la generazione della traiettoria e l'ottimizzazione non lineare. La prima è usata per generare una nuova traiettoria mentre la seconda prende in ingresso una traiettoria già esistente e la modifica per ridurre al minimo il costo.

4.1.2. Generazione della traiettoria

La generazione di una nuova traiettoria è basata su un'implementazione di A^* , un algoritmo di attraversamento grafi e ricerca percorsi.

Il grafo è inizialmente composto dai soli nodi di partenza e obiettivo per poi essere generato dinamicamente durante l'esecuzione, espandendo i nodi esistenti: ogni volta che un nodo viene visitato, un numero predefinito di nodi figli viene generato e aggiunto ai nodi aperti del grafo.

I nuovi nodi sono generati usando un passo fisso e un insieme di angoli di virata congruenti tra loro. Il numero di successori da generare ad ogni iterazione è deciso arbitrariamente, così come il massimo angolo di virata. Un albero è usato per memorizzare ogni nodo in ordine di visita.

Il costo per raggiungere ogni nuovo stato è calcolato e assegnato al nuovo nodo. In seguito al calcolo del costo, A^* stima il costo rimanente per raggiungere il nodo obiettivo. Tale stima prende il nome di euristica.

4.1.2.1. Euristiche in A*

Il calcolo dell'euristica è una fase fondamentale nel processo di ottimizzazione dell'algoritmo.

L'euristica prende in considerazione semplici dati per calcolare, per ogni nodo, una stima del costo rimanente per raggiungere l'obiettivo.

In A*, al fine di ottenere un'euristica ammissibile, la stima deve essere ottimistica, cioè deve rappresentare un valore certamente più basso del costo reale per raggiungere l'obiettivo dal nodo analizzato.

Un'euristica ammissibile permette di espandere prima i nodi più promettenti, modificando la lista di priorità che regola l'ordine di espansione dei nodi. Allo stesso tempo, è possibile interrompere un percorso non promettente prima di quanto sia altrimenti possibile.

Tale lista prende il nome di frangia, la quale memorizza tutti i nodi ancora aperti. La frangia è ordinata secondo la somma del costo per raggiungere un nodo e l'euristica ad esso associata.

L'algoritmo termina quando un nodo appena creato raggiunge il nodo finale. L'euristica incide nell'ordine di espansione al fine di anticipare la terminazione dell'algoritmo, riducendo la necessità di espandere nodi in direzioni poco promettenti.

4.1.2.2. Modello di calcolo dell'euristica

Il modello usato per calcolare l'euristica in questa implementazione è basato sulla creazione di una configurazione fittizia, che, a partire dalla configurazione reale di stato, recettori e nodo finale, permette di calcolare una stima ottimistica del costo richiesto per raggiungere il nodo finale.

Nella configurazione fittizia, le direzioni del nodo finale e del velivolo vengono allineate in modo da permettere di percorrere in linea retta la distanza che li separa. Di seguito, i recettori sensibili vengono posti tutti alle spalle del velivolo mantenendo la stessa distanza da esso alla quale si trovano nella configurazione reale. Da qui, l'euristica assume come valore il costo risultante dal campionamento del percorso descritto dalla linea retta che va dal nodo analizzato al nodo finale.

Il modello cerca di ricreare le circostanze migliori in relazione alla minimizzazione del costo di una data configurazione nodo-recettori-obiettivo.

Viene minimizzata la distanza da ricoprire per raggiungere l'obiettivo da un dato nodo mentre viene massimizzato il rateo di allontanamento del nodo da tutti i recettori.

Per cui, il valore così calcolato dal modello è sicuramente più basso del costo minimo reale per raggiungere l'obiettivo.

4.1.3. Ottimizzazione della rotta

La fase di ottimizzazione fa uso della programmazione non lineare per minimizzare il costo di traiettorie già formate.

La programmazione non lineare è un processo utilizzato per trovare un ottimo in un insieme non numerabile di possibili soluzioni.

L'implementazione utilizza l'algoritmo *Sequential Least Squares Programming (SLSQP)* (Python, s.d.).

4.1.3.1. Il processo

Il processo inizia suddividendo la traiettoria in segmenti, le cui giunzioni diventano dei punti di passaggio per la traiettoria. Tali punti possono essere traslati su un asse di traslazione.

Gli assi di traslazione sono linee passanti per i punti sopra delineati e perpendicolari alla traiettoria già esistente.

La traiettoria può traslare di un certo *delta* sugli assi, il cui valore massimo è definito a priori.

Al fine di generare delle traiettorie percorribili, sono imposti dei limiti alle differenze di *delta* consecutivi ed è imposto un massimo angolo di curvatura tra due segmenti adiacenti.

L'obiettivo dell'ottimizzazione è di minimizzare il costo della traiettoria, inteso come la somma dei segmenti che la compongono.

4.1.3.2. Minimi locali

La presenza di minimi locali nella funzione di costo crea la possibilità di trovare una soluzione non ottima dato un percorso preesistente.

Per ridurre tale rischio, vengono create diverse configurazioni di partenza che vengono poi ottimizzate e delle quali viene selezionato il percorso di costo minimo.

La creazione delle configurazioni di partenza parte dall'identificazione di due pivot che suddividono la traiettoria in tre segmenti uguali. I pivot individuano due punti di passaggio e sono localizzati approssimativamente ad un terzo e a due terzi della lunghezza del percorso.

Vengono individuate cinque posizioni differenti per ciascun pivot, le quali sono successivamente combinate per generare venticinque configurazioni di partenza totali. Ciò diminuisce le probabilità che il percorso trovato cada in un minimo locale, trovando una soluzione non ottima.

4.1.4. Considerazioni

Da una prima analisi del lavoro allo stato attuale risultano problemi sia dal punto di vista concettuale che prestazionale.

4.1.4.1. Geometria utilizzata

In primo luogo, il modello non tiene in considerazione l'altezza del velivolo nella descrizione del suo stato, che è per cui considerato sempre al livello del suolo. È intuibile che il rumore percepito al suolo, dove sono situati i recettori sensibili, è invece fortemente influenzato dall'altitudine del velivolo.

Ciò causa un comportamento errato dell'algoritmo per quello che riguarda la scelta dei percorsi da espandere.

Percorsi localizzati ad un'altitudine maggiore preferirebbero l'espansione di nodi che minimizzano la lunghezza del percorso rimanente, dirigendosi in maniera più diretta verso l'obiettivo, mentre percorsi più prossimi al suolo tenderebbero a circumnavigare la zona dove la densità di recettori sensibili è maggiore.

L'assenza di una componente verticale genera, inoltre, una singolarità matematica in prossimità dei recettori che rende pressoché impossibile, per i velivoli, transitare nell'area limitrofa.

Ciò avviene per via del costo che tende all'infinito associato ai punti che azzerano la distanza orizzontale con i recettori, non tenendo conto della distanza verticale invece presente.

Un secondo problema che inficia la geometria del contesto analizzato è la definizione che viene fatta della distanza minima tra due punti, basata sulla distanza cartesiana.

La distanza minima tra due punti deve tenere conto, invece, del raggio minimo di curvatura, proprio della modalità con cui il velivolo si muove nello spazio, e della direzione del punto iniziale e finale.

In diversi punti dell'algoritmo, l'assenza di uno studio più approfondito della distanza minima tra due punti provoca problematiche che si riflettono sia sull'efficienza che sull'efficacia dell'algoritmo stesso.

4.1.4.2. Impatto dei recettori

Quando parliamo di inquinamento acustico, prendiamo in considerazione dimensioni che coprono ordini di grandezza molto diversi tra loro. Se l'apporto sul rumore totale di un recettore sensibile molto distante è pressoché trascurabile, il tempo impiegato per calcolarne il costo risulta invariato rispetto alla distanza.

Per questo motivo, al momento, le performance risentono molto della crescita del numero dei recettori considerati, anche se il costo totale è calcolato tenendo prevalentemente conto dei soli recettori più prossimi al velivolo.

4.1.4.3. L'impatto dell'euristica

Il modello di euristica utilizzato modifica di molto le prestazioni temporali dell'algoritmo e risulta avere un effetto anche sul costo dell'effettivo percorso minimo trovato.

Il modello di euristica attualmente utilizzato, oltre a risultare molto pesante dal punto di vista prestazionale, non offre l'apporto preventivato per quanto riguarda il taglio dei nodi visitati nelle aree meno promettenti dell'area di ricerca. Questo lavoro si prefissa come obiettivo di analizzare maggiormente l'impatto che l'euristica ha sull'efficacia e efficienza dell'algoritmo, analizzando il suo apporto e arrivando a proporre una nuova soluzione che risponda ai problemi individuati in tal senso.

Nell'euristica trova, in primo luogo, uno degli effetti che l'utilizzo della distanza cartesiana come distanza minima ha sull'algoritmo.

Non tenere in considerazione il raggio minimo di curvatura del velivolo, può portare a tratte conclusioni sbagliate sull'effettiva vicinanza del nodo obiettivo, diminuendo di molto il valore dell'euristica dei nodi che hanno, in realtà, ancora molta strada da percorrere per raggiungerlo.

Un'ulteriore criticità è data dagli stati che, già distanti, si allontanano dal nodo obiettivo.

L'euristica non ha modo di tenere conto di tale allontanamento, con il risultato che vengono favoriti nodi che si dirigono verso le zone più esterne dell'area di ricerca. Ciò comporta la necessità di prostrarre i percorsi aperti per molto in aree poco ottimali, aumentando il tempo di esecuzione dell'algoritmo.

Il risultato è un'euristica associata ai nodi molto spesso poco proporzionata, ad un costo computazionale invece molto alto da sostenere.

4.1.4.4. Prossimi passi

Nei capitoli successivi troverà spazio un'analisi più approfondita delle prestazioni dell'algoritmo, sia dal punto di vista temporale che per quanto riguarda la correttezza del risultato generato.

A partire da ciò, verranno proposte delle modifiche incentrate al miglioramento dell'algoritmo da entrambi i punti di vista, le quali saranno poi testate e, eventualmente, integrate nell'implementazione finale.

4.2. Analisi dello stato attuale

In questo capitolo è presentata un'analisi delle prestazioni attuali dell'algoritmo. Tale analisi verterà sullo studio dell'algoritmo generativo attualmente basato su A^* e su come le approssimazioni e l'euristica utilizzata modifichino sostanzialmente il tempo di ricerca del percorso minimo, nonché la qualità della soluzione stessa.

4.2.1. Velocità dell'algoritmo generativo

La prima fase dell'ottimizzazione della rotta prevede la generazione di un percorso tramite l'utilizzo dell'algoritmo di ricerca A^* .

L'implementazione utilizza una generazione dinamica del grafo usato per la ricerca del percorso minimo.

La generazione procede aggiungendo al grafo dei nodi protraendo quelli che vengono visitati, approssimando il movimento del velivolo.

Infatti, poiché il velivolo si muove in uno spazio continuo si è scelto, al fine di salvaguardare le prestazioni dell'algoritmo, di approssimare il movimento considerando degli stati finiti occupati dal velivolo nel tempo.

In questa fase, la velocità con la quale l'algoritmo trova una soluzione è strettamente legata alla lunghezza del passo e al numero di nodi, oltre che all'euristica utilizzata.

Inoltre, per effetti causati dall'aumento nel tempo dei nodi considerati, il rateo di iterazioni al secondo che l'algoritmo è in grado di effettuare ha un carattere decrescente.

Il numero di nodi influenza, in primis, il tempo necessario a gestire le strutture ordinate che li contengono.

Nella ricerca bidirezionale tale carattere decrescente è peggiorato dalla necessità di controllare, per ogni nodo generato, la giunzione con l'albero opposto, processo che richiede di controllare la sovrapposizione tra il nodo e i nodi appartenenti all'intero albero opposto.

Nei paragrafi successivi verrà analizzato il tempo necessario a trovare una soluzione modificando queste variabili.

È obiettivo di questo capitolo utilizzare tale analisi al fine di migliorare le prestazioni temporali dell'algoritmo.

4.2.2. Comparazione delle prestazioni tra A* e algoritmo di Dijkstra

L'algoritmo A* impiega un'euristica basata su un calcolo ottimistico del costo da un nodo n al nodo destinazione per cercare di limitare l'esplorazione del grafo.

La stima della lunghezza di un cammino dall'origine alla destinazione passante per un nodo n è data dalla (4.2).

$$f(n) = g(n) + h(n) \quad (4.2)$$

$g(n)$ rappresenta il costo del cammino dal nodo origine a n e $h(n)$ è la stima euristica per difetto della lunghezza da n alla destinazione. L'introduzione dell'euristica agisce sulla stima del costo aumentandone il valore. Possiamo considerare l'algoritmo di Dijkstra come un caso particolare di A* nella quale vale la (4.3).

$$h(n) = 0 \quad (4.3)$$

Nei paragrafi successivi è presentata un'analisi del numero di nodi generati e del tempo di esecuzione per A* e l'algoritmo di Dijkstra.

4.2.2.1. Numero di nodi visitati

Aumentare il valore dell'euristica assegnata ad ogni nodo ha come obiettivo principale quello di modificare l'ordine di espansione dei nodi, preferendo quelli più promettenti.

Ciò si traduce, teoricamente, nella riduzione del numero di nodi che è necessario esplorare prima di decretare un percorso minimo.

La [Tabella 4.2](#) riassume i dati risultati dai test effettuati in rapporto al numero di nodi visitati nel caso di A* e nel caso dell'algoritmo di Dijkstra.

I test sono stati effettuati sull'area rappresentante il circondario dell'aeroporto di Malpensa, a partire dalle due piste di decollo e utilizzando la ricerca bidirezionale.

La [Tabella 4.1](#) riassume le configurazioni utilizzate per i test.

Tabella 4.1: Configurazioni utilizzate per i test

Configurazione	Rotta	Area	Recettori
1	FX7 opt volato	100kmx100km	210
2	FX7 opt volato	60kmx60km	210
3	FX7 opt volato	60kmx60km	216
4	KZ49 long opt	60kmx60km	210
5	KZ49 long opt	60kmx60km	216

Tabella 4.2: Nodi visitati nel caso di A* e dell'algoritmo di Dijkstra su diverse configurazioni

Config.	Tre angoli		Cinque angoli		Sette angoli		Nove angoli	
	A*	Dijkstra	A*	Dijkstra	A*	Dijkstra	A*	Dijkstra
1	831	343	6775	2548	27318	3970	62893	276
2	77	100	491	344	978	1962	2707	232
3	102	163	376	836	1828	3437	1686	530
4	177	201	1117	1387	2818	714	2420	1236
5	49	55	183	205	637	489	1195	957
Media	247,2	172,4	1788,4	1064	6715,8	2114,4	14180,2	646,2

Seppur A* riesca ad eliminare dei nodi nella maggior parte dei casi, mediamente l'algoritmo di Dijkstra visita un numero minore di nodi.

L'incremento di A* rispetto all'algoritmo di Dijkstra si verifica soprattutto nei casi in cui A* ha spazio di espandersi in aree limitrofe dell'area di ricerca, nelle quali, l'implementazione attuale dell'euristica impone di protrarre i percorsi molto più a lungo prima che essi vengano scartati.

Ciò avviene a causa dell'euristica molto alta associata ai nodi che li avvicinano al nodo obiettivo rispetto a quelli che si allontanano dai recettori.

La [Figura 4.1](#) evidenzia un caso in cui tale problematica si presenta: questa fa riferimento alla *configurazione 4* eseguita con nove angoli, caso in cui il numero di nodi è addirittura doppio rispetto all'esecuzione con l'algoritmo di Dijkstra.

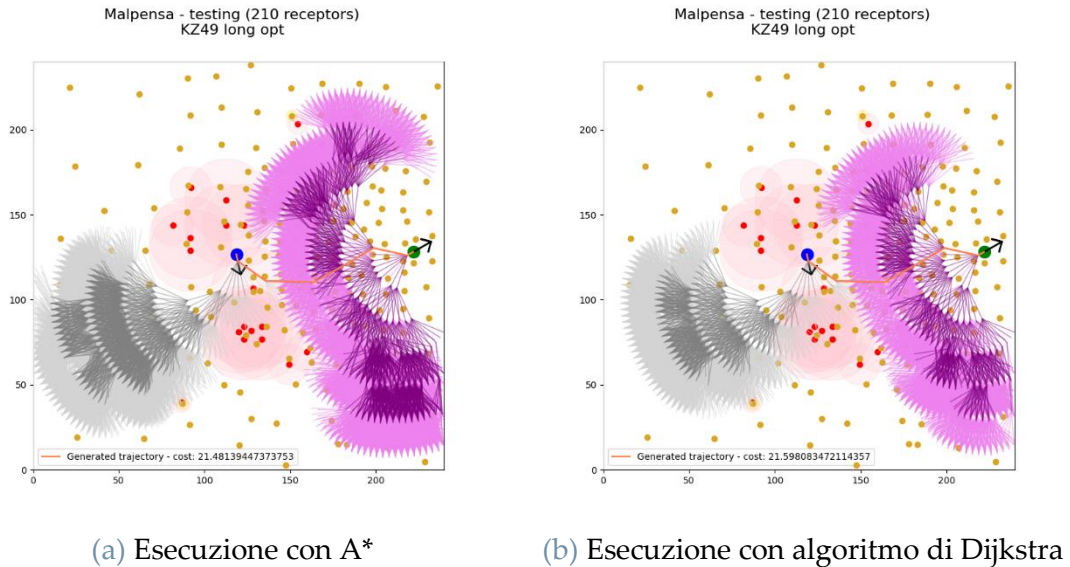


Figura 4.1: Differenza nella generazione dei nodi utilizzando A* e l'algoritmo di Dijkstra nella configurazione 4

Mediamente, il divario tra il numero di nodi dei due algoritmi risulta più evidente all'aumentare degli angoli considerati per ogni espansione.

È da notare come, nel caso dell'algoritmo di Dijkstra, l'esecuzione con nove nodi risulti, in termini di nodi visitati, oltre che più conveniente rispetto all'utilizzo di A*, anche più conveniente rispetto alle esecuzioni con lo stesso algoritmo di Dijkstra sviluppando meno nodi per iterazione.

4.2.2.2. Tempo di esecuzione

Seppur l'euristica di A* ha tra gli obiettivi la riduzione del tempo di esecuzione, questo non si verifica nel caso studiato.

La Tabella 4.3 riassume i risultati delle prove dal punto di vista del tempo di esecuzione.

Le configurazioni utilizzate sono le stesse riportate in Tabella 4.1. Dai test si evidenzia che il tempo impiegato per eseguire l'algoritmo A*, allo stato attuale, è di diverse volte superiore all'esecuzione dell'algoritmo di Dijkstra, cioè ponendo il valore dell'euristica pari a zero per ogni nodo.

Anche quando il numero di nodi visitati dall'algoritmo A* necessario a trovare una soluzione è minore rispetto all'algoritmo di Dijkstra, il tempo di esecuzione è invece molto maggiore.

Tabella 4.3: Confronto dei tempi necessari a raggiungere una soluzione nel caso di A* e dell'algoritmo di Dijkstra

[s]	Tre angoli		Cinque angoli		Sette angoli		Nove angoli	
	A*	Dijkstra	A*	Dijkstra	A*	Dijkstra	A*	Dijkstra
Config 1	23,27	1,19	391,85	19,03	2811,85	42,42	10838,24	2,63
Config 2	1,30	0,25	12,79	1,30	36,57	12,27	139,17	2,02
Config 3	1,70	0,41	11,60	4,49	89,25	24,96	103,59	4,75
Config 4	3,65	0,54	45,45	6,99	180,60	5,39	177,64	12,29
Config 5	1,09	0,16	7,06	1,18	36,87	3,58	93,87	9,44
Media	6,2	0,51	93,75	6,6	631,03	17,72	2270,5	6,23
Media nodi/s	39,87	338,039	19,076	161,21	10,643	119,323	6,25	103,724

La causa è da ritrovarsi nell'eccessiva quantità di calcoli effettuati per ottenere il valore dell'euristica di ogni nodo, che non è giustificata dalla ridotta ed eventuale diminuzione del numero degli stessi.

L'euristica, come nel caso del costo di un segmento, è calcolata tramite la somma del campionamento di diversi punti equidistanti del segmento da percorrere. Ad ogni iterazione, $g(n)$ è calcolato come il costo del passo verso il nodo successivo sommato a $g(n - 1)$.

È quindi sufficiente campionare un segmento di lunghezza relativamente corta e sommarlo al costo del nodo precedente.

Invece, al fine di calcolare $h(n)$, è necessario campionare un segmento di lunghezza pari al percorso più breve che porta dal nodo all'obiettivo.

Tale percorso è mediamente molto più lungo rispetto alla lunghezza del passo. Ciò comporta un forte aumento del tempo medio necessario alla generazione di ogni nuovo nodo rispetto al caso con $h(n)$ nullo.

Il rateo di nodi visitati al secondo passa da un range di 6-40 dell'implementazione attuale con A*, a 100-340 nel caso in cui l'euristica venga posta a zero, come evidenziato nella Tabella 4.3.

4.2.2.3. Valore dell'euristica

Una seconda problematica si riscontra quando si parla delle direzioni in cui si espande il grafo.

In [Figura 4.2](#) è possibile notare come, anche in presenza dell'euristica, il grafo si espanda indistintamente in ogni direzione possibile prima di arrivare a conclusione.

L'insieme dei nodi esplorati da A^* differisce di poco da quelli esplorati dall'algoritmo di Dijkstra se non per i nodi più avanzati appartenenti agli alberi di ricerca, che nel primo caso risultano assenti.

Allo stato attuale, in breve, non si riesce a bloccare precocemente percorsi poco promettenti.

È, invece, necessario prostrarli per molto prima che l'aggiunta dell'euristica abbia un effetto.

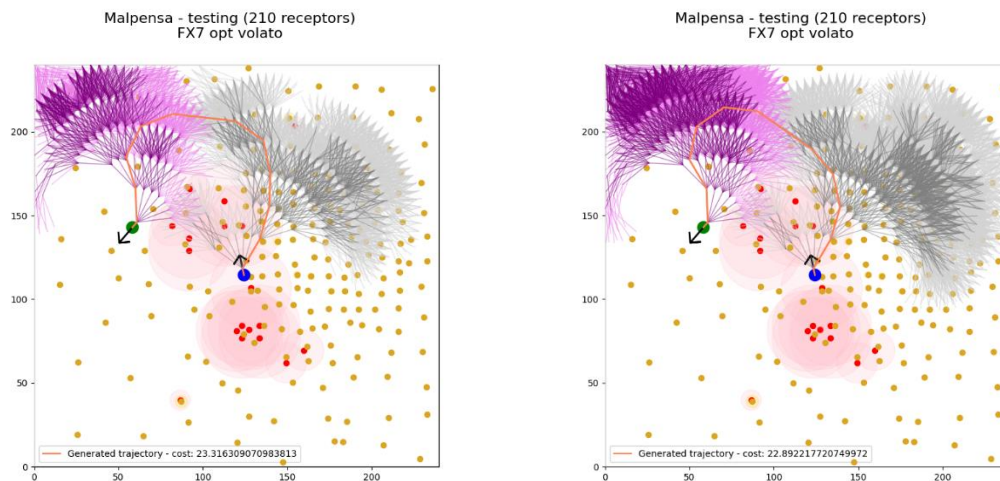
Il problema sta nel fatto che il valore generato dall'attuale modalità di calcolo dell'euristica è molto contenuto, rappresentando esso un caso ideale. Dalle prove effettuate, l'euristica rappresenta all'incirca il valore di un passo. Ciò vuol dire che, con A^* , il percorso viene scartato in media un passo prima rispetto a quanto verrebbe fatto con l'algoritmo di Dijkstra.

Ottenere un valore per l'euristica risulta, quindi, molto costoso da calcolare per i motivi precedentemente citati, e ha un impatto minimo sull'espansione del grafo. Il tempo impiegato per arrivare a sviluppare il percorso risulta quindi di gran lunga maggiore, con A^* in quanto il guadagno prestazionale non riesce a coprire i costi dell'introduzione dell'euristica calcolata alla maniera attuale e, quindi, rendendo l'algoritmo di Dijkstra più conveniente dal punto di vista temporale rispetto ad A^* in questo caso, pur generando più nodi nel processo.

Il metodo di calcolo basato sull'allontanamento costante porta anche altri due svantaggi:

- Assegna un valore più basso ai nodi che sono già lontani dai recettori. Questa situazione si verifica spesso in zone che risultano poco significative per il raggiungimento dell'obiettivo e che, però, fanno fatica ad essere scartate se non dopo diverse iterazioni.
È importante notare come, essendo questi nodi spesso abbastanza avanzati in termini di profondità di espansione, il loro numero è molto elevato a causa del carattere esponenziale della generazione del grafo.
- Durante il calcolo dell'euristica, i nodi che partono distanti dai recettori raggiungono distanze dai recettori impossibili da far verificare nella configurazione reale.

Ciò comporta un apporto dell'euristica molto basso, che avvantaggia così nodi in cui è meno probabile si debba passare per raggiungere l'obiettivo.



(a) Albero generato con A*

(b) Albero generato con algoritmo di Dijkstra

Figura 4.2: Un confronto tra gli alberi di ricerca generati da A* e dall'algoritmo di Dijkstra

4.2.2.4. Utilizzo dell'algoritmo di Dijkstra

L'utilizzo dell'algoritmo di Dijkstra offre, allo stato attuale, prestazioni più soddisfacenti ma potrebbe risultare meno efficiente nelle configurazioni più complesse.

L'aumento del numero di nodi visitati, di molto maggiore rispetto all'esecuzione con A* in alcune configurazioni, impattano in breve tempo sulle prestazioni medie finali dell'algoritmo.

Infatti, il rateo di nodi visitati al secondo assume un carattere decrescente all'aumentare del numero dei nodi stessi, a causa dei calcoli supplementari dovuti al maggior numero di interazioni tra loro.

4.2.3. Problema sulla giunzione tra i due alberi nella ricerca bidirezionale

Osservando i percorsi generati restringendo gradualmente l'area di ricerca, si possono notare delle forti variazioni nella forma dei percorsi generati i quali, talvolta, possono assumere una forma sensibilmente diversa.

Questo comportamento si verifica alla stessa maniera al variare del tipo di euristica utilizzata, evidenziando un comportamento problematico dell'algoritmo. Il risultato atteso prevederebbe la generazione di un percorso minimo uguale tra le differenti esecuzioni, in quanto tali variabili non ne dovrebbero modificare l'ottimalità.

Si evidenziano differenze anche tra gli alberi risultanti dall'esecuzione dell'algoritmo utilizzando un'euristica basata sulla distanza cartesiana tra il nodo e l'obiettivo, una basata sulla distanza minima tenendo in conto il massimo raggio di curvatura ed infine ponendo il valore dell'euristica a zero.

A seguire, in [Figura 4.3](#), [Figura 4.4](#) e [Figura 4.5](#) sono presentati degli esempi di esecuzione su un'area totale di 60kmx60km, utilizzando un fattore di cinque nodi per espansione, suddivisi per tipologia di euristica.

Sono state considerate aree di diverse estensioni, dimensionate in modo da poter contenere per intero il percorso minimo risultante dall'esecuzione dell'algoritmo esente da vincoli di area.

È possibile notare che le traiettorie generate riducendo l'area di ricerca varino sensibilmente anche a parità di euristica utilizzata.

Modificando anche di poco l'area di ricerca, allo stato attuale, cambia il percorso minimo trovato, il quale assume costi differenti.

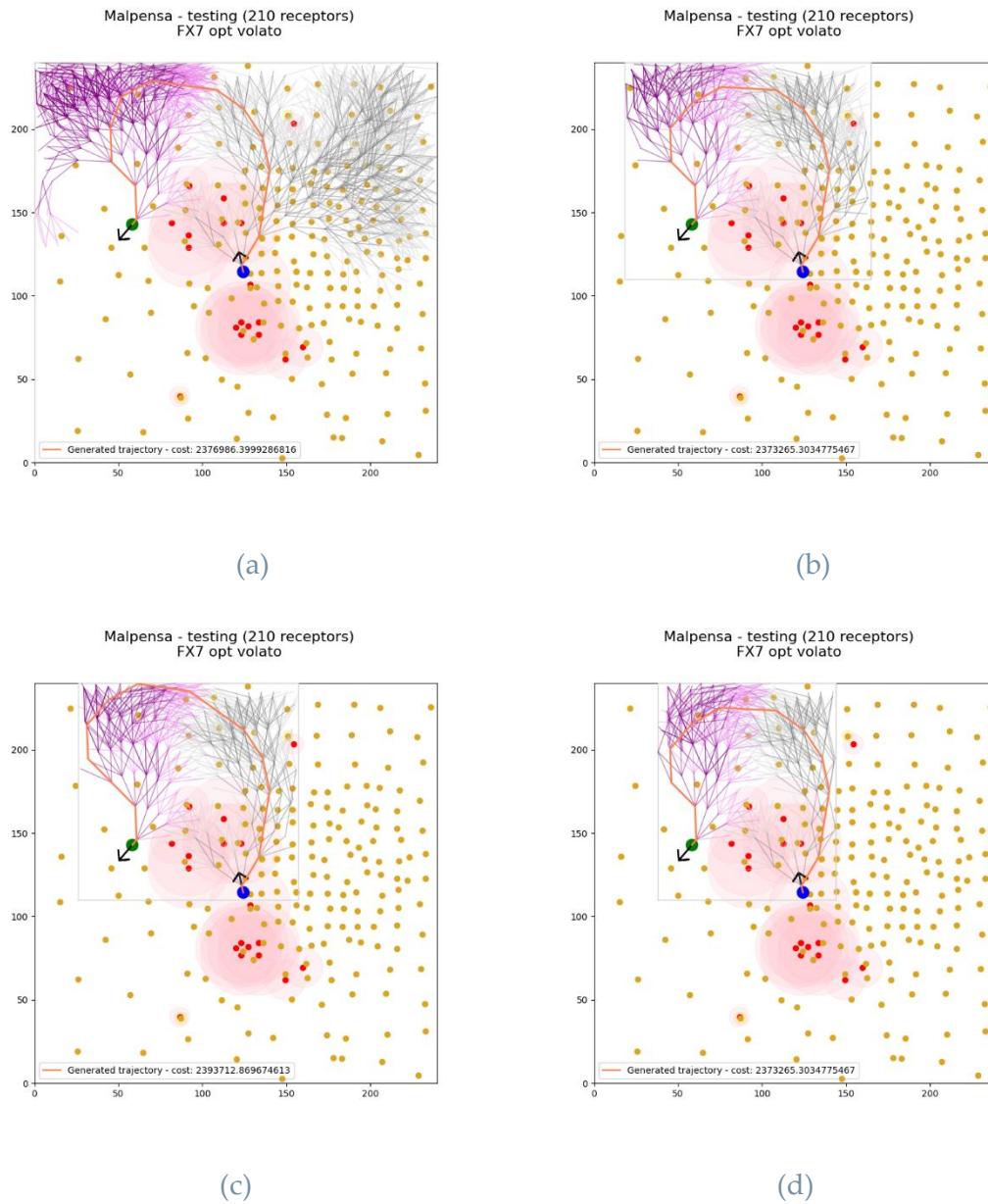


Figura 4.3: Alberi generati con distanza cartesiana come distanza minima

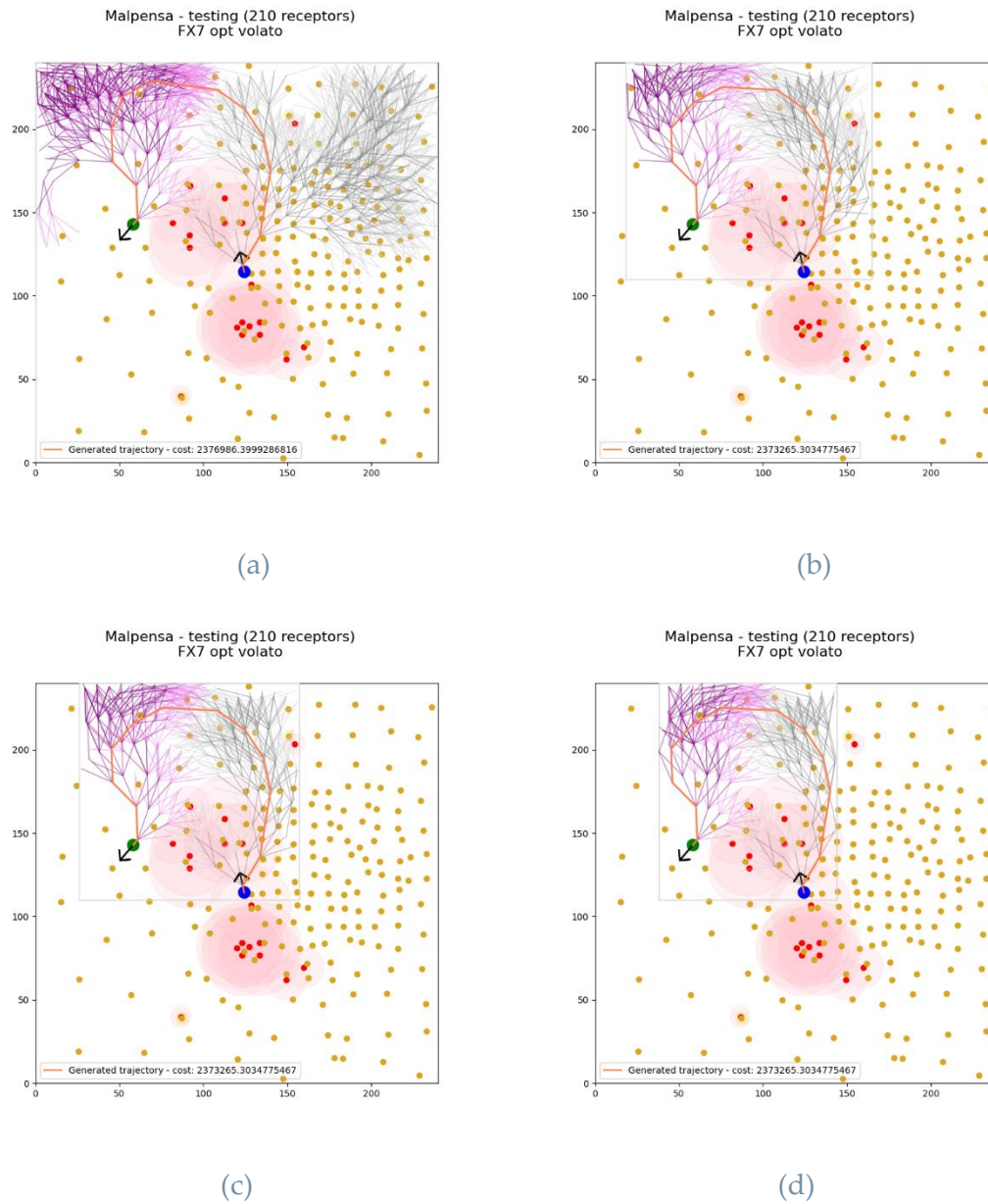
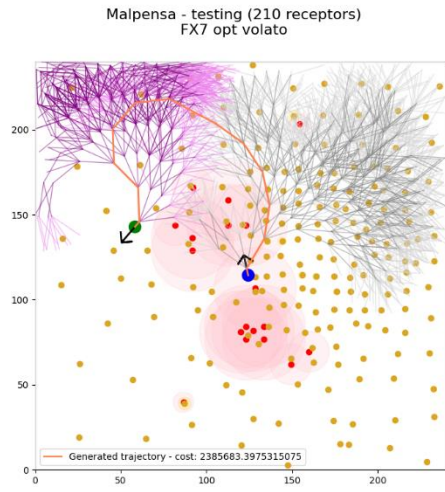
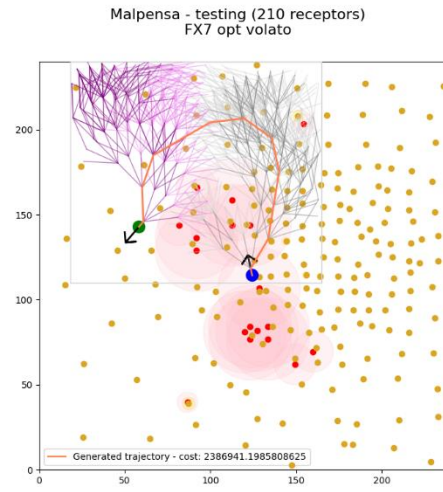


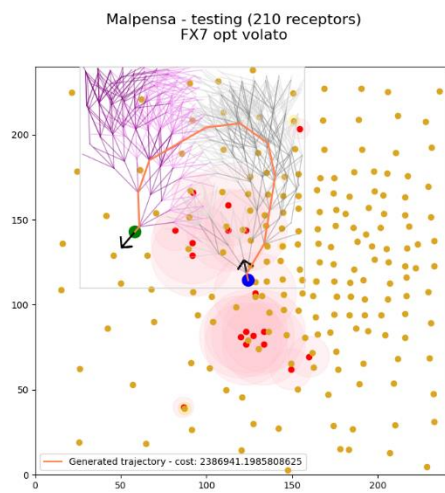
Figura 4.4: Alberi generati con distanza di Dubins come distanza minima



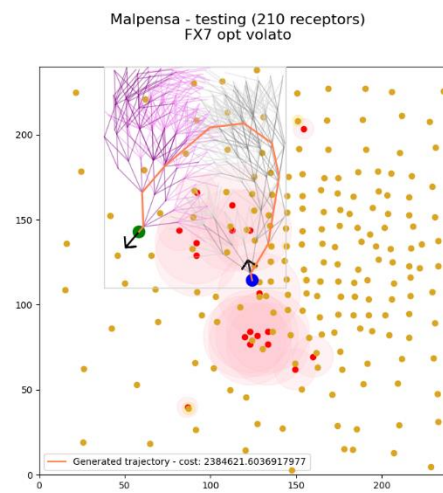
(a)



(b)



(c)



(d)

Figura 4.5: Alberi generati con algoritmo di Dijkstra

4.2.3.1. Il problema delle approssimazioni

I motivi alle spalle di questo problema sono molteplici.

In primis, la generazione dinamica dell'albero e la discretizzazione del passo introducono delle approssimazioni.

Ordine di espansione

L'espansione dei nodi è influenzata dall'euristica utilizzata, la quale varia l'ordine in cui questi vengono visitati e, di conseguenza, la forma degli alberi nel momento in cui si rileva una giunzione.

Al contempo, l'espansione dei nodi differisce per via del fatto che i confini dell'area sono dimensionati in modo da eliminare zone in cui è presente un'alta densità di nodi ma in cui è poco probabile passi il percorso di costo minimo.

Considerando che gli alberi vengono espansi a turno, è probabile che nel momento in cui si rileva una giunzione, uno dei due alberi si trovi in uno stato più avanzato rispetto all'evoluzione che aveva senza limitazioni di area, nella quale venivano precedentemente espansi nodi che non è stato necessario espandere una volta tagliati via, salvando turni di espansione.

Inoltre, i percorsi generati dai due alberi non utilizzano lo stesso insieme di nodi ma anzi, è presente un necessario sfasamento nelle loro coordinate nel caso in cui per quell'area arrivi prima uno o l'altro albero.

Per cui, ad espansioni dei due alberi differenti corrispondono percorsi dai costi differenti.

Discretizzazione del passo

Poiché la traiettoria è originariamente continua, la scelta di discretizzare il passo al fine di ridurre la complessità computazionale genera due ulteriori problemi.

La condizione di terminazione dell'algoritmo bidirezionale prevede che un nodo appena generato si sovrapponga ad un secondo nodo appartenente all'albero dei nodi chiusi opposto.

Considerando che le coordinate che può assumere uno stato sono limitate dalla dimensione fissa del passo, è improbabile che ciò si verifichi.

È stata, quindi, inserita un'approssimazione nel modo di rilevare una giunzione tra due nodi che consiste nella ricerca di nodi appartenenti all'albero opposto entro un dato raggio.

Una volta generato un nodo se ne verifica la distanza dagli altri nodi opposti, prevedendo anche un controllo delle direzioni dei due stati congiunti.

Al fine di evitare che l'espansione scavalchi il nodo dell'albero opposto, il raggio di ricerca deve essere maggiore o uguale alla lunghezza del passo.

Un divario così ampio potrebbe portare l'algoritmo a terminare in modo eccessivamente precoce, aumentando l'incertezza sulla correttezza della soluzione trovata.

Inoltre, il divario tra i due alberi viene riempito da un segmento di lunghezza variabile che entra a far parte a tutti gli effetti nella soluzione generata.

Il costo dello stesso è imprevedibile poiché il segmento non fa parte di nessuno dei due alberi e genera una approssimazione ulteriore, essendo, in più, calcolato in seguito alla terminazione dell'algoritmo.

Il vantaggio in termini di tempo di esecuzione dato dalla ricerca binaria, in questo caso, si paga in termini di correttezza del risultato.

Date le approssimazioni utilizzate, non è possibile avere un'esecuzione sempre coerente tramite la ricerca bidirezionale.

4.2.3.2. Il problema nella definizione della giunzione

Al fine di ridurre l'aleatorietà dovuta al segmento di giunzione tra i due alberi, in seguito al rilevamento di una giunzione, viene espanso nuovamente il nodo padre e vengono controllate le eventuali giunzioni degli altri nodi figlio.

Viene quindi scelto il percorso di costo minore tra quelli generati in questa fase.

Tale processo è necessario, inoltre, poiché la rilevazione di una giunzione è effettuata prima che l'espansione di un nodo sia completa.

Vanno, per cui, controllate le giunzioni di tutti i nodi generati dal nodo padre.

Questo processo ha però un problema di implementazione.

Prima di tutto, vengono generati i nodi figli a partire dal nodo padre del nodo in cui si rileva la giunzione e alla raccolta di tutte le giunzioni relative.

Viene poi calcolato il costo di ogni percorso trovato come somma di costo del nodo dell'albero opposto e costo tra nodo candidato e nodo opposto.

Il costo del nodo padre è uguale per ognuno dei nodi che si stanno considerando ma non viene considerato nel calcolo il costo del segmento che collega il nodo padre e nodo figlio.

Tale segmento, però, varia a seconda del nodo figlio e va considerato per definire il percorso meno costoso.

Viene quindi valutato e selezionato il percorso minore.

In seguito, viene creato il percorso minimo come somma dei due emi-percorsi che vanno, uno, dall'obiettivo al nodo appartenente all'albero opposto per cui si trova la nuova giunzione e, l'altro, dal nodo iniziale al nodo per cui si è trovata la prima giunzione.

Ciò rappresenta un errore in quanto il percorso minimo, a seguito della scelta della nuova giunzione, passa per il nodo figlio per cui tale giunzione è rilevata.

4.2.3.3. Lunghezza del passo e raggio di ricerca della giunzione

Al fine di ridurre l'incertezza introdotta dal segmento di giunzione, sarebbe possibile ridurre il raggio di ricerca della giunzione.

Dalle prove effettuate, però, riducendo tale raggio al 5% della lunghezza del passo, seppur i risultati sono stati meno variabili rispetto alle prove precedenti, in media sono stati trovati dei percorsi più costosi.

Potrebbe essere una soluzione ridurre, invece, la lunghezza del passo stesso ad una frazione di quella utilizzata al momento.

Ciò comporta, però, un aumento esponenziale del numero di nodi visitati.

Tale modifica andrebbe, quindi, accompagnata ad un'euristica che risulti più performante sia dal punto di vista della velocità di calcolo che dell'apporto dato alla riduzione del numero di nodi visitati, essendo essa fondamentale per tagliare in fretta rami non utili.

Le caratteristiche dell'euristica sono al momento incompatibili con l'obiettivo di ridurre l'espansione nelle zone poco promettenti e a compensare l'aumento del numero di nodi eventualmente scaturito dalla riduzione della lunghezza del passo.

4.2.4. Problemi minori

Quanto trattato in questo paragrafo risulta concettualmente problematico, pur non inficiando le performance dell'algoritmo.

4.2.4.1. Coefficiente di bias sulla distanza

L'implementazione attuale dell'algoritmo prevede la presenza di un bias verso percorsi più brevi che ha l'obiettivo di poter ridurre la lunghezza dei percorsi generati, dando un peso maggiore ai percorsi proporzionalmente alla lunghezza stessa.

Tale bias è introdotto tramite una modifica del valore associato al costo del percorso e dell'euristica, introducendo un coefficiente b_1 e un esponente b_2 , secondo le formule (4.4) e (4.5).

$$g(n)_{bias} = g(n) + b_1 * lunghezza\ percorso^{b_2} \quad (4.4)$$

$$h(n)_{bias} = h(n) + b_1 * distanza\ goal^{b_2} \quad (4.5)$$

Nell'implementazione del calcolo del costo percorso risulta, però, un problema. Essendo il costo calcolato secondo la formula (4.6), cioè come il costo del nodo precedente sommato al costo del segmento che congiunge i due nodi consecutivi, la lunghezza percorso considerata dal bias è quella di quest'ultimo segmento.

$$g(n) = g(n - 1) + c(n, n - 1) \quad (4.6)$$

Ciò trasforma la componente esponenziale in una costante, in quanto la lunghezza del passo è fissa per ogni nodo.

Viene a mancare, quindi, l'obiettivo iniziale di dare un peso esponenzialmente maggiore all'aumentare della distanza del percorso.

Inoltre, il valore dell'euristica varia, invece, esponenzialmente con la distanza che separa il nodo dall'obiettivo.

Considerato ciò, l'utilizzo dell'esponente potrebbe rendere non ammissibile l'euristica così modificata, in quanto essa assumerebbe un valore maggiore del costo del percorso rimanente.

Ciò non si verifica con il coefficiente b_1 che, per la sua caratteristica di linearità, non inficia l'ammissibilità dell'euristica risultando in una soluzione, seppur approssimativa, possibile.

Al fine di rendere l'implementazione del bias possibile, si può tenere conto della distanza totale del percorso, sommando la distanza percorsa e la distanza minima da percorrere per raggiungere l'obiettivo dal nodo.

Invece che modificare il costo del percorso e l'euristica separatamente, sarebbe possibile inserire una componente di bias nella funzione di costo che tenga conto di entrambe.

La funzione di costo risultante risulterebbe nella (4.7).

$$f(n) = g(n) + h(n) + b_1 * (L(s, n) + D(n, g))^{b_2} \quad (4.7)$$

Dove $L(s, n)$ è la lunghezza del percorso tra il nodo iniziale e il nodo n e $D(n, g)$ è la distanza di minima tra il nodo n e il nodo destinazione.

Questa modifica permetterebbe di prendere in considerazione la distanza rimanente nel processo di selezione del percorso meno costoso ma non permette di rapportarla al costo in maniera diretta.

Potrebbe risultare più utile costruire un'euristica che stimi il costo necessario per raggiungere l'obiettivo in base alla distanza.

4.2.4.2. Controllo di giunzione

Attualmente, il criterio di terminazione dell'algoritmo prevede che, in seguito all'espansione di un nodo, un nodo figlio da esso generato si trovi o nei pressi dell'obiettivo, nel caso della ricerca monodirezionale, oppure nei pressi di un nodo appartenente all'albero opposto, nel caso di quella bidirezionale.

Tale implementazione del controllo di giunzione può causare problemi da diversi punti di vista.

In primis, dato il passo discreto effettuato dai nodi dopo un'espansione, è altamente improbabile che il nodo figlio si collochi nello stesso punto della griglia rispetto all'obiettivo.

La soluzione adottata per la risoluzione di questo problema consiste nel ricercare il nodo obiettivo entro un certo raggio, una volta generato un nodo, oppure uno o più nodi dell'albero opposto.

Inoltre, si controlla che i due nodi siano direzionati nello stesso senso, entro un certo intervallo.

Questa soluzione non assicura che la traiettoria generata sia percorribile ed impone che il raggio utilizzato per la ricerca sia di lunghezza maggiore o uguale del passo dell'algoritmo di ricerca.

Ciò può, a seconda dei casi, risultare problematico.

Un raggio troppo ampio può risultare in una terminazione precoce dell'algoritmo: decretare la sovrapposizione di due nodi troppo distanti tra loro non assicura che il percorso trovato sia il minimo.

Ridurre il raggio ad una lunghezza minore di un passo, invece, può portare ad oltrepassare il nodo obiettivo non rilevando la sovrapposizione.

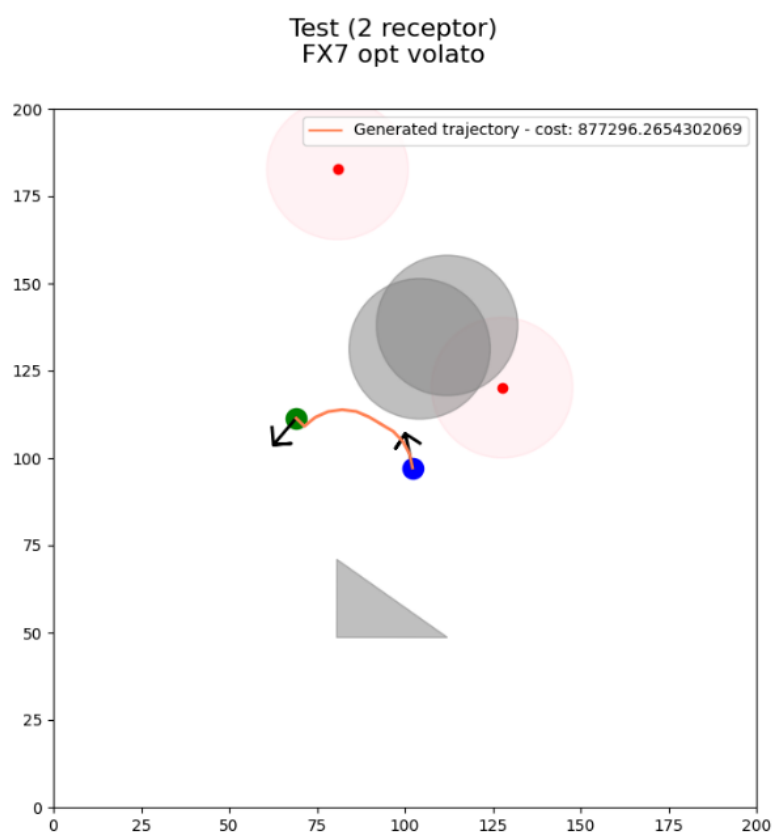


Figura 4.6: Percorso generato dall'algoritmo in cui è presente una terminazione errata causata dalla funzione di controllo di giunzione

Dalla [Figura 4.6](#) si nota come il percorso non sia percorribile infranti in prossimità della giunzione con il nodo obiettivo.

Nel caso in questione, però, i due criteri di giunzione sono rispettati: il nodo obiettivo è situato entro il raggio di ricerca e i due nodi hanno direzioni paragonabili, ma che il percorso risultante non è percorribile.

Il criterio di giunzione dovrebbe, perciò, tenere conto del raggio minimo di curvatura e delle direzioni di entrambi i nodi.

5 Il nuovo approccio

In questo capitolo vengono proposte delle nuove idee che modificano la struttura di alcuni componenti dell'algoritmo.

5.1. Distanza minima tra due nodi

A partire da una analisi sullo spostamento del velivolo nello spazio è emersa la necessità di definire il comportamento di quest'ultimo nel caso della minimizzazione della distanza tra due stati.

In particolare, si è voluto osservare come un velivolo possa raggiungere da un generico stato A, caratterizzato da una certa direzione e coppia di coordinate, un altro stato B, caratterizzato da direzione e coordinate differenti, percorrendo una distanza che risulti minima.

Tale distanza è inficiata da un vincolo sul massimo raggio di curvatura che non permette di arrivare dallo stato A allo stato B seguendo il percorso tracciato dalla distanza cartesiana tra i due.

5.1.1. Percorsi di Dubins

Il caso in questione è stato ampiamente studiato nel lavoro di L. E. Dubins (Dubins, 1957), che dimostra l'esistenza di una possibile geodetica, cioè una curva che descrive il cammino più breve su un dato spazio.

Nel caso in questione la geodetica può occupare soli sei percorsi in presenza di un set (u, U, v, V, R) , caratterizzato da due punti u e v , due direzioni ad essi collegate U e V e di un raggio di curvatura che sia in ogni caso maggiore o uguale di un numero fisso R .

Tali percorsi sono caratterizzati da, al più, tre elementi geometrici e sono suddivisi in due classi, *CSC* e *CCC*, composte come in (5.1) e (5.2).

$$\mathbf{CSC} = \{\mathbf{LSR}, \mathbf{RLS}, \mathbf{LSL}, \mathbf{RSR}\} \quad (5.1)$$

$$\mathbf{CCC} = \{\mathbf{LRL}, \mathbf{RLR}\} \quad (5.2)$$

C rappresenta un arco di circonferenza di raggio R , che può essere direzionata verso sinistra L o destra R , mentre S rappresenta una linea retta.

L'ordine nella stringa rappresenta l'ordine di apparizione nel percorso descritto. La Figura 5.1 mostra i sei percorsi, nelle loro possibili declinazioni.

Nella prima classe, le circonferenze sono localizzate in modo da risultare passanti per i punti u e v e rispettivamente tangenti alle direzioni U e V .

Su tali circonferenze sono situati i due archi che partono dalla posizione del velivolo e arrivano al punto di tangenza tra le due circonferenze, prese a coppie, e una retta.

Il segmento che congiunge i due raggi di circonferenza descritti sopra è posizionato su tale retta, tra i due punti di tangenza.

Nella seconda classe, a collegare le due circonferenze è una terza circonferenza.

Un percorso composto da tre circonferenze è una possibile geodetica in casi in cui i due stati di partenza e arrivo sono distanti tra loro al massimo di due volte il raggio R . (Shkel, 1996)

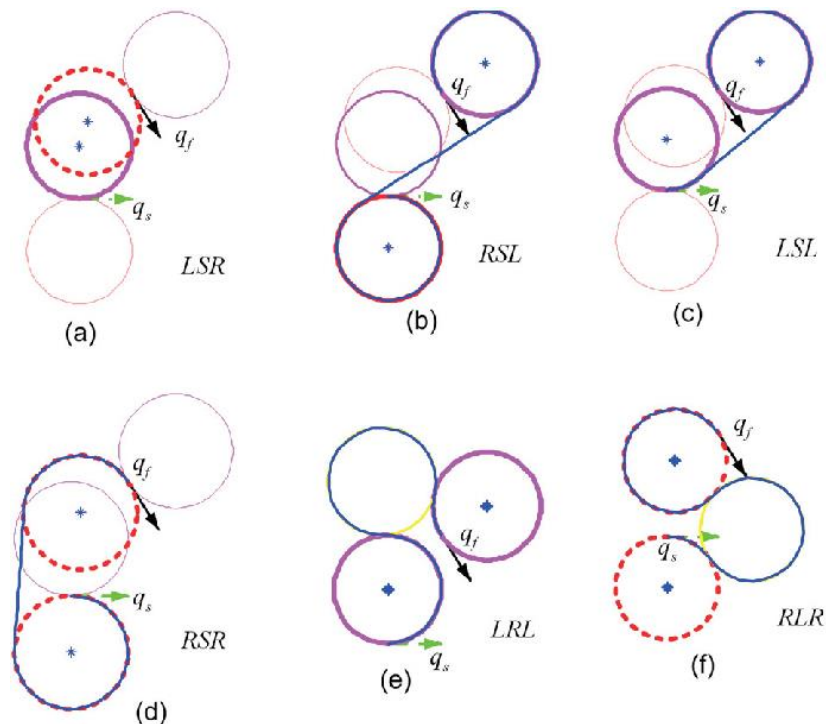


Figura 5.1: I sei possibili percorsi di Dubins da un punto iniziale q_s a un punto finale q_f

5.1.2. Implementazione dei percorsi

I percorsi di Dubins sono stati integrati nel codice tramite l'implementazione di LaValle (LaValle, 2006), che permette di calcolare la lunghezza dei percorsi in tempo reale, scegliendone successivamente il più corto.

Questa implementazione abilita, per la sua velocità, la possibilità di utilizzare i percorsi per migliorare concettualmente e prestazionalmente alcune fasi dell'algoritmo.

5.1.3. Applicazione dei percorsi di Dubins

5.1.3.1. Euristiche

Un primo impiego dei percorsi di Dubins trova spazio nel computo del valore dell'euristica.

Come precedentemente citato, l'euristica, per ogni nodo visitato, è calcolata strutturando un caso ideale in cui il velivolo va a ricoprire la distanza minima che va dal nuovo nodo al nodo obiettivo, allontanandosi dai recettori come se li avesse tutti alle spalle.

Il concetto di distanza minima utilizzato fin ora è stato quello di distanza cartesiana tra i due nodi ma, come intuibile, in presenza di vincoli imposti dalle direzioni dei nodi e dal raggio minimo di curvatura, non rappresenta necessariamente la distanza più breve che li separa.

In presenza di due nodi direzionati in maniera opposta, ad esempio, la distanza minima risulta sensibilmente maggiore della distanza cartesiana.

L'incremento maggiore, in percentuale, si ottiene quando i due nodi sono localizzati a breve distanza cartesiana l'uno dall'altro ma direzionati in modo non consecutivo, il che non permette un percorso diretto dal nodo iniziale a quello obiettivo.

La distanza dei percorsi di Dubins risulta in ogni caso maggiore della distanza cartesiana.

Aumentare la distanza su cui l'euristica è calcolata ne aumenta, di conseguenza, il valore.

L'aumento del valore permette, in primis, che l'euristica abbia una valenza maggiore se rapportata a $g(n)$ e secondariamente, che l'apporto dell'euristica sia maggiormente proporzionato alla distanza effettiva rimanente da percorrere per raggiungere il nodo obiettivo.

Ciò apporta, teoricamente, una diminuzione dei nodi che è necessario visitare per raggiungere una soluzione.

In particolare, potrebbe essere necessario protrarre meno volte i nodi che vanno in direzioni non utili ai fini della generazione di un percorso fattibile.

5.1.3.2. Controllo di sovrapposizione tra due nodi

Un secondo impiego dei percorsi trova riscontro nel controllo della sovrapposizione tra due nodi.

Come visto nel [Paragrafo 4.4.2](#), controllare se due nodi sono sovrapposti, in questo contesto, risulta problematico ma necessario per la terminazione dell'algoritmo. Non è possibile decretare una sovrapposizione controllando esclusivamente un raggio e un angolo fisso senza incorrere in casistiche in cui il percorso risulti impossibile da percorrere in quanto l'angolo formato dalle direzioni dovrebbe variare in base alla distanza tra i due nodi.

Una via alternativa prevedrebbe di calcolare la lunghezza del percorso di Dubins tra due nodi di cui si vuole controllare la sovrapposizione e verificarne la lunghezza. Se questa risultasse inferiore ad una certa soglia, se ne può decretare la sovrapposizione.

Nell'implementazione attuale, il raggio di ricerca è uguale alla lunghezza di un passo per evitare che il nodo, con un passo successivo, scavalchi l'altro senza rilevarne la vicinanza.

Con i percorsi di Dubins, mantenendo la stessa soglia di approssimazione, si ottiene un miglioramento dal punto di vista concettuale, in quanto si decretano sovrapponibili due nodi la cui direzione e posizione relativa sono compatibili con una traiettoria percorribile.

Dal punto di vista computazionale, calcolare il percorso di Dubins richiede uno sforzo maggiore rispetto al calcolo della distanza cartesiana. Ciò non dovrebbe risultare, però, in un costo computazionale eccessivo nel caso della ricerca monodirezionale ma potrebbe nella ricerca bidirezionale.

Se nel primo caso, infatti, per ogni nuovo nodo si deve controllare la distanza dal solo nodo obiettivo, nel secondo si effettua un controllo per ogni nodo chiuso appartenente all'albero di ricerca opposto.

5.2. Introduzione terza dimensione

Attualmente, la modalità di movimento del velivolo è caratterizzata da un passo di lunghezza fissa e direzione variabile sul piano, che rappresenta la superficie terrestre.

Sullo stesso piano sono localizzati i recettori sensibili che rappresentano le città e la densità abitativa.

Non è invece considerata una variazione d'altitudine del velivolo che, trovandosi in contesto di decollo, dovrebbe essere tenuta in conto durante l'ottimizzazione.

L'idea più immediata è quella di modificare l'espansione dei nodi in modo da introdurre altitudini differenti su cui generare i nuovi nodi.

Questa modalità d'introduzione della terza dimensione nel caso in oggetto risulta in diverse problematiche.

In primo luogo, la complessità di calcolo dell'algoritmo cresce esponenzialmente con il numero di livelli di altezza differente considerati ad ogni passo, a causa del grado di libertà ulteriore dalla terza dimensione.

Siano k il numero di passi necessari a raggiungere l'obiettivo, n il numero di angoli considerati ad ogni passo e m il numero di livelli di altezza considerati ad ogni espansione, la complessità della versione tridimensionale dell'algoritmo è $O(k^{n * m})$.

In secondo luogo, l'altitudine è da considerare anche nel calcolo del costo e dell'euristica relativo ad ogni nuovo nodo.

Essa ne modifica il valore in quanto cambia la distanza del velivolo da ogni recettore.

A parità di coordinate geografiche, tale distanza aumenta con l'altitudine del velivolo.

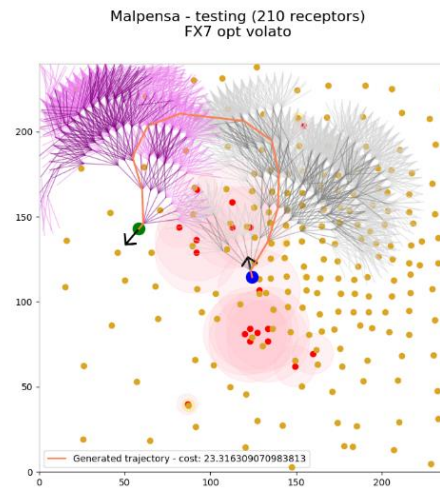
Al contempo, l'introduzione della terza dimensione apporta sostanziali miglioramenti alla soluzione trovata.

Nel modello bidimensionale utilizzato fin ora, essendo l'altitudine pari a zero, quando la traiettoria del velivolo passa al di sopra o nei pressi delle coordinate di un recettore sensibile, l'apporto sul costo dato dalla distanza tra quest'ultimo e il velivolo tende all'infinito.

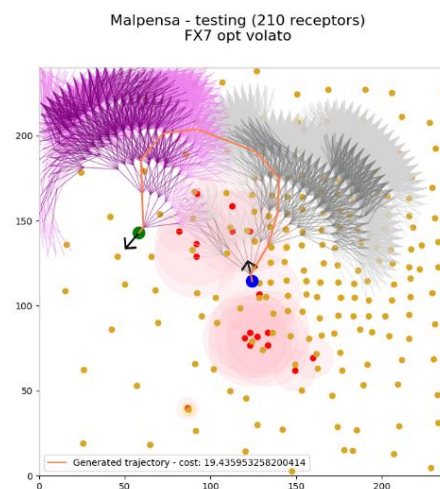
Questa discontinuità matematica viene risolta dall'introduzione della terza dimensione nel caso in cui il velivolo abbia un'altitudine diversa da zero. Anche con coordinate del velivolo uguali a quelle di un recettore, l'altitudine rende la distanza sempre maggiore di zero.

Bisogna inoltre tenere in considerazione che il percorso migliore varia di molto in base all'altezza del velivolo.

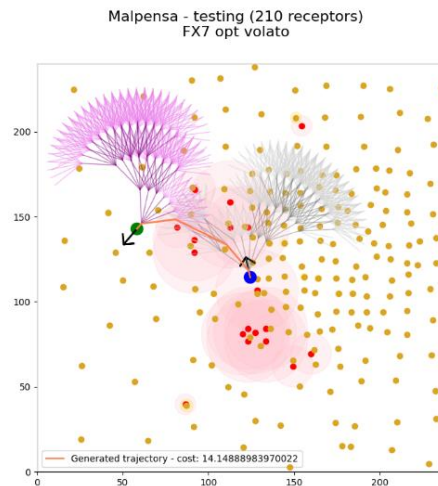
Essendo i recettori tutti al suolo, è intuibile, e dimostrato dai risultati computazionali, che il percorso migliore di un velivolo al suolo tenderà ad aggirare i recettori mentre un velivolo ad altezza di crociera riuscirà ad avvicinarsi in maniera più diretta all'obiettivo, sorvolando i recettori tenendosene comunque a distanza sfruttando l'altitudine.



(a) Altitudine 0m



(b) Altitudine 1500m



(c) Altitudine 3000m

Figura 5.2: Percorsi generati sulla stessa configurazione ma ad altitudini differenti

5.2.1. Crescita costante

Al fine di aggirare l'aumento della complessità dovuto all'introduzione di più livelli sarebbe possibile sfruttare la posizione dei recettori.

Se teniamo in considerazione che questi ultimi sono posizionati tutti sul suolo, è intuibile che all'aumentare dell'altitudine del velivolo corrisponde una decrescita del rumore percepito su di essi dovuto al conseguente aumento della distanza. È, per questo, inutile considerare i percorsi che vanno verso il basso o che, più in particolare, non aumentino la loro quota del massimo consentito dai vincoli imposti. Inoltre, un aumento di quota costante si trova in accordo con il regime di decollo del velivolo, che quindi risulterebbe nell'ottimizzazione di un percorso non dissimile al caso reale.

Date tali considerazioni, la generazione di nodi su più livelli diventa pressoché inutile oltre che sconveniente.

Nel caso venissero generati anche i nodi con altezze minori dalla massima altezza, a parità di coordinate, non farebbero mai parte della soluzione in quanto aventi una distanza minore dai recettori sensibili al suolo.

Tale modalità di implementazione genera lo stesso numero di nodi della ricerca bidimensionale, mantenendone invariata la complessità computazionale, pur ottenendo lo stesso risultato della ricerca tridimensionale.

5.2.2. Problemi implementativi

L'introduzione della terza dimensione porta con sé diverse complicazioni.

La prima si riscontra nell'utilizzo della ricerca bidirezionale. Essa prevede l'utilizzo di due alberi di ricerca, il primo ha origine nel nodo start e procede verso il nodo obiettivo.

Un secondo albero viene generato procedendo a ritroso dal nodo obiettivo verso il nodo iniziale.

In questo caso, le direzioni dei nodi vengono invertite.

Al fine di decretare un percorso come minimo è sufficiente che i due alberi, formati dai nodi chiusi, si incontrino.

Nel caso in esame, però, non è definita a priori l'altezza finale del nodo obiettivo. Questa risulta impossibile da prevedere sia nella versione base dell'implementazione della terza dimensione sia nella versione con incremento costante, nella quale risulta legata alla lunghezza del percorso.

Non è, quindi, possibile individuare un nodo finale da cui far partire l'albero di ricerca inverso.

Una possibile soluzione vedrebbe l'attribuzione al nodo finale di una altitudine arbitraria da cui far partire l'albero inverso.

Il percorso generato da tale imposizione sarebbe, però, inficiato dal nuovo vincolo. Quest'ultimo imporrebbe una lunghezza minima al percorso generato, necessaria a soddisfare l'obbligo di raggiungere tale altezza.

Nel caso della crescita costante, la lunghezza dei percorsi generati sarebbe necessariamente fissa.

Una seconda via vedrebbe l'utilizzo di un'altitudine costante, anch'essa definita in modo arbitrario, per i nodi dell'albero inverso.

Si manterrebbe la crescita costante, invece, per l'albero diretto verso il nodo obiettivo.

In questo caso, si avrebbe l'aggiunta di un passo successivo che prevede, in seguito al riconoscimento di una giunzione tra i due alberi di ricerca, di cambiare le altitudini dei nodi appartenenti all'albero inverso in modo da proseguire la crescita costante dell'albero diretto verso l'obiettivo.

Sarebbe necessario basare il riconoscimento della giunzione esclusivamente sulle coordinate bidimensionali, escludendo l'altitudine.

Un controllo sul rumore di picco ammissibile si rivelerebbe necessario per i nodi dell'albero inverso.

Il percorso generato soffrirebbe, però, dell'approssimazione dovuta alla forma che può essere più o meno diretta verso l'obiettivo a seconda dell'altitudine a cui ci si trova, come mostrato nei paragrafi precedenti.

È per questo bene che i nodi appartenenti agli alberi di ricerca rispecchino l'altitudine finale del velivolo nel momento stesso in cui essi vengono espansi.

Nel caso specifico, molto probabilmente il percorso inverso avente l'altitudine massima avrebbe la tendenza a dirigersi direttamente verso il nodo finale rispetto a quanto farebbe ad un'altitudine minore, ottenendo un risultato particolarmente svantaggioso una volta effettuati gli aggiustamenti necessari per ottenere la crescita costante.

I percorsi trovati da tali soluzioni, visto l'impiego di approssimazioni nel processo di ricerca, potrebbero generare dei percorsi non effettivamente minimi.

Date queste considerazioni e, non essendo necessario avere un risultato in tempo reale, potrebbe essere più vantaggioso utilizzare la ricerca monodirezionale in contesto di crescita costante.

Ciò assicurerebbe l'ottenimento di una soluzione più realistica rispetto al modello bidimensionale evitando di sacrificare, però, la correttezza del risultato utilizzando approssimazioni necessarie per ottenere un miglioramento prestazionale.

L'innegabile vantaggio prestazionale dovuto alla ricerca bidirezionale sarebbe perciò lasciato a sviluppi futuri.

5.3. Taglio rumore con passa-alto

Allo stato attuale, il costo computazionale necessario a raggiungere una soluzione dipende linearmente dal numero di recettori sensibili presenti nella configurazione studiata.

I recettori sensibili non hanno tutti, però, lo stesso impatto nel computo della funzione di costo.

L'impatto dei recettori sensibili decresce quadraticamente rispetto alla distanza, oltre che con il peso proprio del recettore.

Studiando come varia tale impatto in rapporto alla distanza, sarebbe possibile tenere in considerazione i recettori sensibili entro un certo raggio dalla posizione del velivolo, corrispondente alla distanza necessaria a raggiungere una data soglia di intensità sonora.

L'intensità sonora potrebbe essere basata sul livello di rumore già normalmente presente in una città, prendendo come riferimento l'intensità media di una città poco rumorosa.

Ciò avrebbe lo scopo di ridurre l'impatto computazionale causato dal numero di recettori non tenendo in conto quelli non sufficientemente significativi.

La distanza necessaria è calcolata come di seguito.

Il livello dell'intensità sonora è espresso in decibel dalla formula (5.3).

$$L_i = 10 \log_{10} \frac{I}{I_{ref}} = 10dB \quad (5.3)$$

I rappresenta l'intensità sonora, esprime la potenza sonora attraverso una superficie.

La formula (5.4) e (5.5) esprimono il valore di I e I_{ref} .

$$I = \frac{N}{4\pi r^2} \quad (5.4)$$

$$I_{ref} = 10^{-12} \frac{W}{m^2} \quad (5.5)$$

N rappresenta la potenza del motore.

5.3.1. Modello di rumore attuale

5.3.1.1. Motore turbojet

Come riferimento possiamo utilizzare la potenza di un motore turbojet, come nel lavoro di tesi precedente, che corrisponde a 100kW.

Dalla formula (5.3), al fine di raggiungere un livello di intensità sonora pari a 50dB, è necessaria una distanza dal velivolo di circa 282,1 km.

Per raggiungere un livello di 60dB è necessaria una distanza di 89,2km.

Per raggiungere un livello di 70dB è necessaria una distanza di 28,2km.

I risultati sono riportati nella [Tabella 5.1](#).

Tabella 5.1: Distanza alla quale si raggiunge una certa intensità sonora con un motore turbojet

Intensità sonora [dB]	Distanza [km]
50	282,09
60	89,20
70	28,21

Le città italiane hanno una media registrata di intensità sonora che varia tra i 50dB e i 65dB (N. van Oosten, 2009).

Da ciò risulta che, anche prendendo in considerazione una soglia abbastanza alta come 60dB e un'area di ricerca ampia come 100km, si eliminerebbero dal conteggio un numero esiguo di recettori e solo in poche configurazioni.

In particolare, è necessario che il velivolo si trovi nei pressi dei lati più esterni dell'area di ricerca e che dei recettori si trovino al lato opposto della stessa. Tendenzialmente è possibile notare come, nel nostro caso, i recettori sono invece situati nella zona centrale dell'area di ricerca.

Inoltre, i lati dell'area sono tendenzialmente poco interessanti ai fini della ricerca del percorso migliore, che si concentra prevalentemente nella zona centrale. C'è il rischio che, con tale modifica, si avvantaggino leggermente diramazioni esterne poco utili, addirittura aumentando il tempo di esecuzione. Riducendo invece la soglia a 50dB, il numero di recettori eliminati dal conteggio sarebbe nullo per ogni area di ricerca minore di circa 200km per lato.

È da notare come una soglia di 50dB risulterebbe più in linea con l'idea di eliminare dal calcolo i rumori in ogni caso confondibili con il rumore cittadino.

Date queste considerazioni, prendere in considerazione di implementare la modifica, pur potendo produrre vantaggi computazionali, non risulterebbe molto conveniente.

5.3.1.2. Motore turbofan

Dei risultati totalmente differenti si ottengono modificando la scelta del motore da studiare.

Nel caso di un motore turbofan al decollo, la potenza è pari a 1000W. Motori del genere sono più assimilabili alla categoria di velivoli che normalmente decollano da Malpensa.

Di seguito è riportata una tabella che indica a che distanza si raggiunge una data intensità sonora, a partire dalla formula (5.3).

Tabella 5.2: Distanza alla quale si raggiunge una certa intensità sonora con un motore turbofan

Intensità sonora [dB]	Distanza [km]
30	282,09
40	89,20
50	28,21
60	8,92
70	2,82

Le figure sottostanti riportano, in rosso, l'area in cui i recettori sensibili vengono considerati nel calcolo nel caso di una intensità minima di 50dB. L'intensità corrisponde ad un raggio di 28,21km.

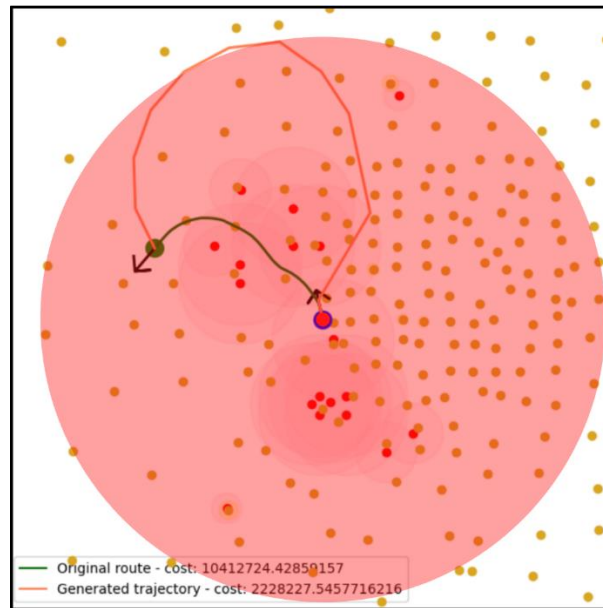


Figura 5.3: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso a

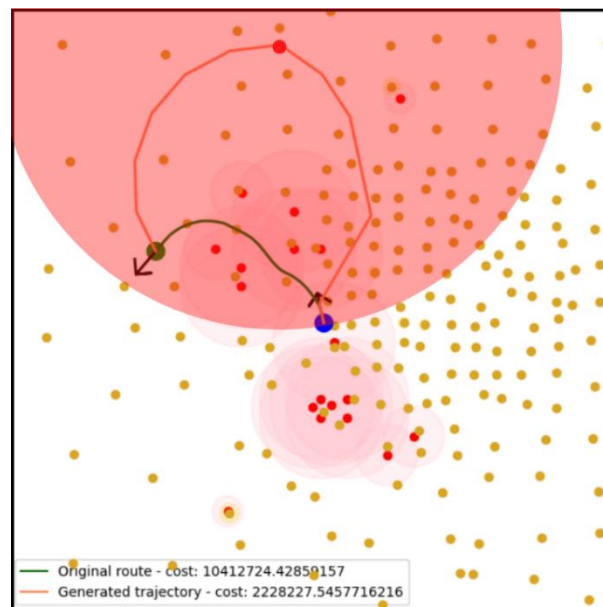


Figura 5.4: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso b

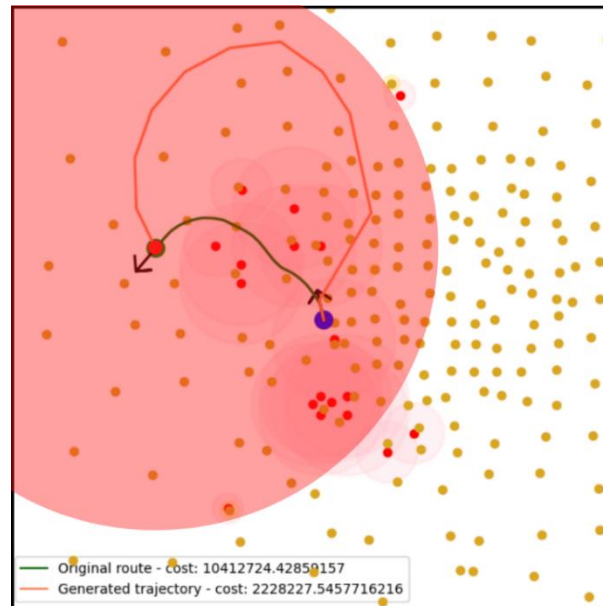


Figura 5.5: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso c

Sono considerate, di seguito, tre diverse posizioni del velivolo:

- La Figura 5.3 mostra il caso di un velivolo posto al centro dell'area. Il vantaggio in questo caso si riduce all'esclusione di alcuni recettori rappresentanti la densità abitativa posti nella regione più esterna del grafico.
- La Figura 5.4 mostra il caso di un velivolo nel punto più esterno del percorso generato. In questo caso il vantaggio è maggiore, riuscendo ad escludere per intero i recettori della parte inferiore del grafico e alcuni nella parte superiore destra.
- La Figura 5.5 mostra il caso di un velivolo nel nodo destinazione: pur non escludendo molti recettori principali, è qui evidente come il vantaggio computazionale può risultare maggiore via via che il velivolo occupa stati più esterni.

È importante notare che esistono due classi di recettori sensibili. I recettori sensibili in rosso rappresentano i comuni localizzati nei pressi dell'aeroporto di partenza.

I recettori gialli sono, invece, una rappresentazione della densità abitativa, generati a partire dei dati del *Global Human Settlement Layer (GHSL)*. (Schiavina, 2019) Il peso di questi ultimi, nel calcolo del costo, è una frazione dei recettori sensibili principali.

Pur non avendo lo stesso peso dal punto di vista del costo, le due tipologie di recettori impattano allo stesso modo sulle prestazioni dell'algoritmo. Se ha senso considerare la densità abitativa a distanze ridotte, l'apporto in termini di costo al crescere della distanza risulta totalmente trascurabile.

Considerato ciò, l'inserimento di un passa-alto con una soglia di 50dB potrebbe risultare in un buon vantaggio prestazionale, riuscendo ad eliminare dal conteggio i recettori appartenenti ad una porzione sostanziale dell'area di ricerca.

L'eliminazione dei rumori al di sotto di 50dB potrebbe però risultare in un cambio della soluzione trovata.

Devono essere effettuate delle prove al fine di controllare quanto varia il costo in relazione alla soglia scelta.

È necessario trovare un valore della soglia che non modifichi eccessivamente il costo finale del percorso ma che renda l'introduzione della soglia vantaggiosa dal punto di vista della complessità temporale dell'algoritmo.

È da notare come, con l'attuale formula del rumore, una soglia di 40dB su un'area di ricerca da 60kmx60km non riesca ad eliminare nessun recettore sensibile per ogni posizione possibile del velivolo.

In conclusione, per avere un vantaggio computazionale è necessario considerare un set di recettori sensibili considerati vicini costante per l'intero processo di campionamento del segmento, evitando di controllare la distanza punto per punto.

Questa approssimazione è necessaria in quanto, controllare la distanza per ogni punto di campionamento, richiederebbe lo stesso sforzo computazionale di calcolarne il costo.

5.3.2. Modello di rumore direzionale

I dati risultanti dalla generazione del modello di rumore basato sulla direzione, descritti nel lavoro di P. Zepeda (Zepeda, 2022), suggeriscono delle distanze necessarie al raggiungimento della soglia minima di 40dB molto più basse rispetto alla equazione (5.3), basata sul quadrato della distanza, utilizzata fino ad ora.

L'implementazione di tale modello aumenta notevolmente il numero di recettori eliminati tramite il filtro bassa-alto, ottenendo al contempo un miglioramento prestazionale dell'algoritmo.

È però necessario valutare gli effetti dell'inserimento della funzione di rumore direzionale sia sulla forma della rotta generata che sull'effettivo guadagno in termini di tempo.

In particolare, bisogna assicurarsi che vi sia un guadagno prestazionale o che la rotta sia diversa da quella ottenuta con l'attuale implementazione, preferendo la seconda in quanto ottenuta con dati che maggiormente rispecchierebbero il caso reale. I risultati computazionali di tali test saranno presentati nel [Capitolo 6](#).

5.3.3. Sinergia con il calcolo dell'euristica

L'introduzione di un filtro passa-alto potrebbe avere degli effetti evidenti anche sulle prestazioni temporali del calcolo dell'euristica.

Seppur non evidente nel calcolo del costo reale di un segmento, l'eliminazione dell'apporto dei recettori sensibili in base alla distanza potrebbe aiutare a ridurre l'impatto prestazionale del numero dei recettori sulle tempistiche di espansione dei nuovi nodi.

Il calcolo dell'euristica rappresenta, infatti, un caso ideale in cui il velivolo si discosta contemporaneamente da tutti i recettori al massimo rateo possibile, raggiungendo distanze non possibili nella configurazione reale.

Eliminare l'apporto dei recettori più lontani, nella configurazione fittizia, porterebbe ad un calcolo più rapido del valore dell'euristica.

Questa, seppur leggermente più bassa rispetto al valore totale, scegliendo una soglia adatta, risulterebbe comparabile alla precedente euristica ma sarebbe calcolata con un dispendio computazionale minore.

Inoltre, l'implementazione di un valore minimo di rumore percepibile da un recettore permetterebbe di modificare il calcolo dell'euristica nella maniera descritta nel paragrafo, rendendo il suo valore direttamente proporzionale alla distanza rimanente da percorrere.

La sinergia generata potrebbe portare ad una forte diminuzione della complessità temporale dell'algoritmo.

5.4. Modifica dell'euristica

Dall'analisi presentata nel [Paragrafo 4.2](#), allo stato attuale l'apporto dato dall'euristica al progetto risulta controproducente per quanto riguarda i tempi di esecuzione.

Si rende necessaria una modifica dell'attuale modello utilizzato.

Sono percorribili diverse strade per ottenere un miglioramento in questo contesto, le quali cercando di risolvere il problema basandosi su un aspetto diverso dello stesso:

- Semplificare il modello dell'euristica in modo da ridurre il tempo impiegato per calcolarla
- Modificare l'euristica aumentandone il valore, in modo da diminuire il numero di nodi da visitare per arrivare ad una soluzione
- Eliminare l'euristica, passando all'algoritmo di Dijkstra

5.4.1. Aumento della distanza minima tramite percorsi di Dubins

È possibile, mantenendo il modello di euristica attuale, aumentare il valore dell'euristica e renderlo più proporzionato alla distanza reale rimanente da percorrere, preservandone l'ammissibilità.

Utilizzando i percorsi di Dubins, infatti, si può aumentare la distanza su cui costruire la configurazione fittizia.

Attualmente, tale configurazione è costruita utilizzando un segmento di lunghezza pari alla distanza cartesiana tra il nodo di cui si calcola l'euristica e il nodo obiettivo, ponendo poi i recettori alle spalle.

Come presentato nel [Paragrafo 5.1](#), la distanza cartesiana è sicuramente minore o uguale alla distanza minima tra i due nodi, che è invece uguale alla distanza del percorso di Dubins più breve.

Si può quindi utilizzare la lunghezza di quel percorso per aumentare il valore dell'euristica ottenendo una stima ottimistica in quanto è preservata la condizione iniziale di percorrere la distanza minima lasciandosi i recettori esattamente alle spalle.

La modifica punta ad aumentare l'euristica di due tipi di nodi:

- I nodi che pur, risultando vicini al nodo obiettivo, non possono raggiungerlo direttamente a causa dei vincoli imposti dalla direzione;

- I nodi lontani che sono direzionati in modo da allontanarsi dal nodo obiettivo, tenendone in conto il percorso ulteriore necessario per direzionarsi verso l'obiettivo.

L'aumento dell'euristica li rende meno prioritari nella coda di espansione, avvantaggiando nodi più promettenti.

La [Tabella 5.3](#) presenta i risultati computazionali ottenuti a riguardo del numero di nodi visitati, in diverse configurazioni, utilizzando

Tabella 5.3: Numero di nodi visitati da A* utilizzando la distanza cartesiana e di Dubins per la generazione della configurazione fittizia

	Tre angoli		Cinque angoli		Sette angoli		Nove angoli	
	Cart	Dub	Cart	Dub	Cart	Dub	Cart	Dub
Config 1	831	537	6775	4245	27318	15304	62893	27316
Config 2	77	65	491	342	978	566	2707	1319
Config 3	102	76	376	256	1828	1042	1686	1086
Config 4	177	255	1117	2145	2818	740	2420	806
Config 5	49	93	183	199	637	467	1195	997
Media	247,2	205,2	1788,4	1437,4	6715,8	3623,8	14180,2	6304,8

È possibile notare che, in generale, il numero di nodi visitati risulta minore nel caso dell'uso dell'euristica basata sulla distanza di Dubins.

La differenza si accentua con l'aumentare dei nodi generati per ogni espansione.

La [Tabella 5.4](#) mostra i tempi di esecuzione tra le due varianti.

L'incremento di complessità introdotto dalla necessità di campionare un segmento uguale alla lunghezza dei percorsi di Dubins invece che alla distanza cartesiana fa sì che la media di nodi visitati per secondo sia leggermente più bassa nell'implementazione che utilizza tali percorsi.

Risulta però evidente che il risparmio dato dalla riduzione dei nodi visitati copra completamente tale complicazione.

Mediamente, infatti, la variante che utilizza i percorsi di Dubins risulta molto più rapida

Tabella 5.4: Confronto dei tempi necessari a raggiungere una soluzione utilizzando la distanza cartesiana e di Dubins per la generazione della configurazione fittizia

[s]	Tre angoli		Cinque angoli		Sette angoli		Nove angoli	
	Cart	Dub	Cart	Dub	Cart	Dub	Cart	Dub
Config 1	23,27	16,89	391,85	274,18	2811,85	1562,20	10838,24	4340,28
Config 2	1,30	1,23	12,79	10,65	36,57	27,50	139,17	84,20
Config 3	1,70	1,52	11,60	9,27	89,25	53,48	103,59	78,68
Config 4	3,65	5,94	45,45	78,60	180,60	49,60	177,64	70,33
Config 5	1,09	2,38	7,06	8,82	36,87	31,53	93,87	92,06
Media	6,2	5,59	93,75	76,3	631,03	344,86	2270,5	933,11
Media nodi/s	39,87	36,708	19,076	18,84	10,643	10,508	6,25	6,757

La riduzione dei nodi visitati, però, non è in grado di coprire l'incremento di complessità portato dall'introduzione dell'euristica stessa nell'algoritmo.

Seppur viene introdotto un miglioramento, questo non è sufficiente a rendere il tempo di esecuzione minore del tempo necessario, invece, utilizzando l'algoritmo di Dijkstra.

Al fine di rendere l'euristica conveniente è necessario ridurre la complessità.

5.4.2. Cambio di modello

Una seconda opzione prevedrebbe di calcolare l'euristica in un modo completamente diverso dall'attuale implementazione.

Il possibile nuovo modello considera la distanza minima da percorrere dal nodo visitato al nodo obiettivo e la trasforma in una stima del costo tramite un coefficiente che ne permetta di mantenerne l'ammissibilità. $h(n)$ è quindi modificata come nella (5.6).

$Dubins(n, g)$ rappresenta la distanza minima rimanente tra il nodo n e il nodo obiettivo g .

$$h(n) = \text{Dubins}(n, g) * k \quad (5.6)$$

La distanza minima è spesso utilizzata come euristica con A* per la sua immediatezza di calcolo che risulta essere una caratteristica fondamentale, come visto in precedenza.

Al fine di ottenere una euristica ammissibile è, però, necessario utilizzare un coefficiente che non generi un costo superiore al costo minimo effettivo del percorso rimanente.

La modifica prevedrebbe di ottenere tale coefficiente prendendo come riferimento il costo del punto dell'area di ricerca in cui il rumore è minimo, cioè, in cui risulta minima la sommatoria del rumore percepito su ogni recettore sensibile. Il coefficiente risultante è espresso nella (5.8).

Inoltre, bisogna tenere in considerazione che il costo di un segmento è normalmente calcolato in maniera discreta, tramite un processo di campionamento. Il costo, con il campionamento, è calcolato come somma del rumore per ogni punto campionato.

Quindi, lunghezza del passo di campionamento L_{samp} modifica il valore ottenuto. Al fine di ottenere un risultato non dipendente da L_{samp} e, in particolare, rapportabile a $g(n)$, è sufficiente discretizzare anche la distanza di Dubins. In questa istanza, quindi, l'euristica è calcolata come nella (5.7).

$$h(n) = \frac{\text{Dubins}(n, g)}{L_{samp}} * k \quad (5.7)$$

$$k = w(x_{min}, y_{min}) = \min \left(\sum_i \frac{1}{r_i^2} \right) \quad (5.8)$$

Il valore che risulta è sicuramente più basso del costo del percorso minimo reale in quanto, l'euristica così costruita, è equivalente a campionare il percorso più breve come se in ogni punto campionato il velivolo generasse il rumore minimo possibile.

Questa modalità genererebbe un vantaggio sia sulla complessità del calcolo eseguito che sul valore ottenuto, in particolare:

- Risulterebbe molto più rapida dell'attuale implementazione e riuscirebbe ad occupare una piccola percentuale del tempo di generazione di un nuovo nodo: si sostituisce una semplice moltiplicazione, al campionamento di un segmento fittizio di lunghezza variabile e complessità dipendente dal numero di recettori totali.
Calcolare la distanza restante risulta immediato tramite l'implementazione dei percorsi di Dubins mentre il coefficiente è calcolato campionando preventivamente la configurazione una singola volta.
- Il valore ottenuto sarebbe proporzionale alla distanza rimanente per raggiungere l'obiettivo.
Al contrario, l'euristica attualmente utilizzata rapporta il valore alla massimizzazione della distanza dai vari recettori, che non risulta avere una corrispondenza diretta con l'effettiva condizione di terminazione dell'algoritmo.
- Si raggiunge un valore più alto per l'euristica dei nodi periferici, più lontani dai recettori ma anche dal nodo obiettivo.
Si riduce, in questo modo, l'espansione di nodi periferici che si allontanano dall'obiettivo, che l'attuale euristica però avvantaggia.

Con l'aggiunta della terza dimensione, sarebbe necessario modificare il coefficiente utilizzato in base all'altezza propria dello stato di cui si vuole calcolare $h(n)$.

5.4.2.1. Implementazione con soglia minima di rumore

Al fine di ridurre l'impatto che un campionamento di tutta l'area di ricerca avrebbe sul tempo di esecuzione dell'algoritmo, sarebbe possibile eliminare questa fase nell'eventualità di un inserimento di filtro passa-alto per il rumore.

È possibile impostare tale soglia accordandola tramite dei test o utilizzando una soglia per cui l'effetto del rumore sul costo del percorso risulta trascurabile come 40dB.

Tale livello di intensità sonora risulta confondibile con rumori di fondo sul territorio (N. van Oosten, 2009).

Il valore dell'euristica sarebbe quindi calcolato tramite la formula (5.9).

$$h(n) = \text{Dubins}(n, g) * R * I_{min} \quad (5.9)$$

Dove R è il numero di recettori presenti sull'area e I_{min} è uguale alla soglia minima dell'intensità sonora.

Ciò rappresenta, però, un caso ideale in cui, in particolare, il velivolo si trova in ogni momento ad una distanza sufficientemente elevata da risultare fuori dall'area di interesse di ogni recettore.

Il valore dell'euristica potrebbe, per cui, risultare eccessivamente basso e non avere gli effetti sul risparmio temporale sperati.

5.5. Definizione dinamica dell'area di ricerca

Al fine di diminuire il tempo di esecuzione necessario alla terminazione dell'algoritmo, risulta particolarmente efficace ridurre l'area in cui è possibile, per l'algoritmo, ricercare il percorso.

Attualmente, il processo di definizione dell'area è lasciato all'utente che può scegliere se effettuare aggiustamenti manuali.

Dai test effettuati risulta che il percorso minimo trovato generando più nodi per espansione non è di molto dissimile rispetto al percorso generato con soli tre nodi. La [Figura 5.6](#): Percorsi generate sulla stessa configurazione utilizzando un numero diverso di nodi per espansione mostra un esempio dei due percorsi generati sulla stessa configurazione utilizzando rispettivamente tre (a) e nove (b) nodi per espansione.

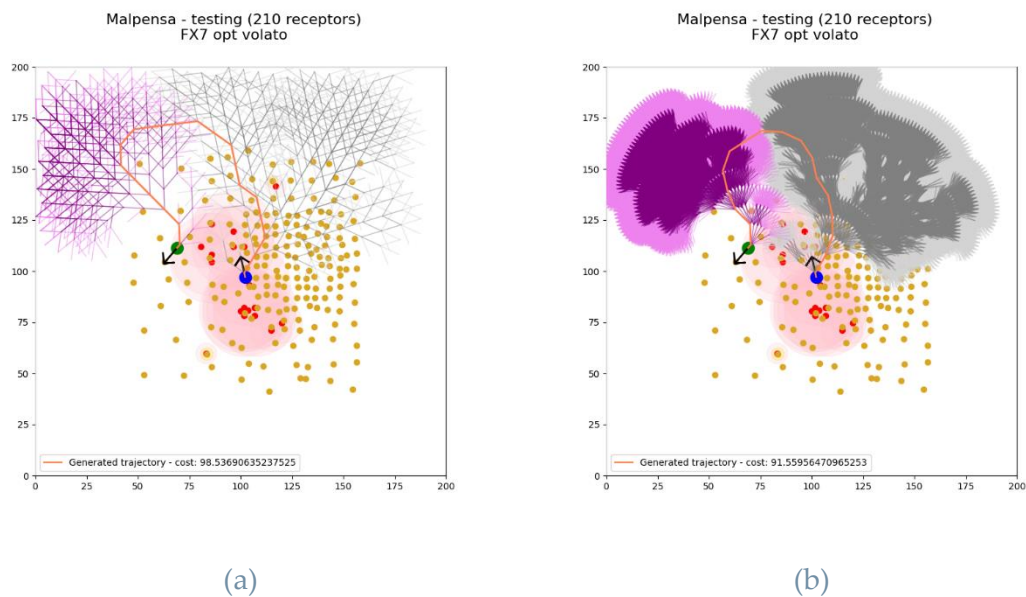


Figura 5.6: Percorsi generate sulla stessa configurazione utilizzando un numero diverso di nodi per espansione

I due percorsi, infatti, sono collocati nella stessa zona dell'area di ricerca. È evidente, inoltre, che al variare del numero di nodi considerati per ogni espansione, il tempo di esecuzione cambia di diversi ordini di grandezza.

Le esecuzioni con tre nuovi nodi per espansione terminano, in media, in 1 – 4 secondi per aree di ricerca da 60kmx60km.

È possibile sfruttare questa caratteristica per diminuire il tempo necessario per trovare una soluzione nei casi che utilizzano molti angoli.

5.5.1. Descrizione del processo

Il processo si configurerebbe in una prima fase caratterizzata da una ricerca a tre nodi, posti in maniera da rappresentare lo stato del velivolo curvando seguendo il raggio massimo verso destra e sinistra e procedendo in avanti.

Questa fase è necessaria al fine di scandagliare velocemente l'area e definirne i contorni.

I contorni sono in seguito utilizzati per definire l'area di ricerca massima nel quale eseguire la ricerca a nove o più nodi.

È necessario offrire un certo margine per evitare che il percorso migliore risultante dall'esecuzione con più angoli risulti più esterno dell'area e venga, di conseguenza, tagliato fuori.

6 Risultati computazionali

6.1. Aeroporto di Malpensa

Il caso preso in esame si riferisce all'aeroporto di Malpensa (IATA: MXP, ICAO: LIMC), in cui sono stati svolti dei test che concernono la procedura di decollo da due piste direzionate in maniera opposta.

L'aeroporto si trova nei comuni di Somma Lombardo (VA) e Ferno (VA), situati nella parte nord-occidentale della Lombardia.

6.1.1. Le piste

L'aeroporto è fornito di due piste parallele, la 35L/17R e la 35R/17L, entrambe orientate a 349° (35L, 35R) o a 169° (17R, 17L) a seconda della direzione di utilizzo. Le piste sono lunghe 3920 m e sono posizionate a 808 m di distanza l'una dall'altra, la prima posizionata più a ovest della seconda.

È possibile percorrere le due piste in entrambi i sensi ma normalmente vengono utilizzate le piste orientate verso nord, lasciando a casi particolari l'utilizzo del senso opposto.

6.1.2. Procedure di abbattimento del rumore attive

Il gruppo SEA, responsabile per la gestione del sistema di aeroporti di Milano, sta portando avanti dei progetti per la salvaguardia dell'ambiente che circonda l'aeroporto di Malpensa con un'attenzione particolare per quanto riguarda la riduzione dell'inquinamento acustico nelle zone limitrofe.

La commissione incaricata della gestione dell'aeroporto ha già adottato misure che prevedono la modifica delle rotte in modo da bilanciare l'impatto acustico sulle diverse aree e garantisce dei periodi di silenzio assoluto ad intervalli regolari su ognuna delle aree sensibili.

Le procedure di decollo sono attualmente al vaglio della commissione al fine di minimizzare l'impatto acustico sulle zone limitrofe.

Le misure di prevenzione del rumore attualmente in uso sono (SEA s.p.a., 2013):

- Limitazioni sul tipo di velivolo, in favore di modelli più moderni
- Definizione di una nuova rotta per il decollo dalla pista 35R/17L
- Uso di rotte P-NAV per rispettare le rotte assegnate
- Schematizzazione delle rotte per distribuire spazialmente e temporalmente in maniera uniforme le operazioni di volo
- Restrizioni operative per il periodo notturno
- Ottimizzazione del profilo verticale in fase di decollo e di atterraggio
- Regolamentazione delle attività di terra

6.2. Hardware e software utilizzato

I test sono stati effettuati utilizzando due versioni del software sviluppato che corrispondono alle versioni prima e dopo l'implementazione delle modifiche e delle considerazioni discusse nei capitoli precedenti.

Il software è stato testato su un Lenovo Thinkpad T14 con le seguenti specifiche:

- Intel i7-1165G7 @ 2.80GHz
- Intel Iris Xe Graphics
- 16 GB, 3200 MHz, single-channel RAM
- Windows 10 Enterprise, versione 20H2

Tutti i tempi di esecuzione fanno riferimento questa configurazione hardware e software.

6.3. Casi studio

Al fine di testare le differenze generate dall'implementazione delle modifiche presentate nel [Capitolo 5](#), verranno considerati due casi studio.

Entrambi i casi sono basati sulla fase di decollo di due rotte, le quali seguono rispettivamente le Standard Instrument Departure (SID) DOGUB5L e SRN5W, visibili nelle carte nautiche presentate in [Figura 6.1](#) e [Figura 6.2](#).

Le traiettorie generate con la nuova implementazione verranno confrontate per costo e forma della stessa, per numero di nodi visitati e per tempi di esecuzione a parità di hardware utilizzato, alle traiettorie generate dalla precedente implementazione, nonché per costo e forma alle rotte attualmente in uso.

La configurazione dell'algoritmo si differenzia dalla precedente per le modifiche presentate nei precedenti capitoli.

In particolare, si fa riferimento con "Configurazione 1" nei casi studio a venire l'algoritmo che utilizza:

- Percorsi di Dubins per calcolo della distanza minima;
- Un incremento di altitudine del velivolo costante del 10%;
- Taglio dei recettori sensibili distanti più di 4000m
- Funzione di rumore basata su distanza e direzione del velivolo
- Utilizzo dell'algoritmo di Dijkstra
- Diminuzione dinamica dell'area di ricerca
- Ricerca monodirezionale

Si fa riferimento con "Configurazione 2" l'algoritmo che utilizza:

- Distanza cartesiana per calcolo della distanza minima
- Altitudine costante a 0m
- Funzione di rumore basata sulla potenza del motore e sulla distanza
- Utilizzo di A*
- Ricerca bidirezionale

6.3.1. Partenza SID DOGUB5L

Il primo caso di studio prevede un decollo dalla pista 35R, direzionata a nord est. Al fine di comparare i risultati con una rotta effettivamente utilizzata sono stati utilizzati dati forniti da Flightradar24, un servizio di tracciamento globale che permette di ottenere informazioni sulle traiettorie di voli in tempo reale.

Per permettere delle comparazioni con la traiettoria generata dall'algoritmo, è stata selezionata la parte iniziale del volo cargo FX7/N865FD diretto a Memphis (MEM), con partenza da Malpensa (MXP) alle 20:11 del 17 Marzo 2021.

La tabella [Tabella 6.1](#): Configurazione nodo iniziale e finale della rotta FX7/N8865FD presenta le coordinate iniziali e finali, con i rispettivi azimuth, del punto di partenza e di arrivo della rotta.

Tabella 6.1: Configurazione nodo iniziale e finale della rotta FX7/N8865FD

	Posizione iniziale	Posizione finale
Latitudine	45.617424°	45.680832°
Longitudine	8.737066°	8.525457°
Azimuth	348°	237°

6.3.1.1. Analisi dei percorsi

Al fine di rapportare i risultati prodotti con quelli ottenuti dalla precedente implementazione, i recettori sensibili sono stati distribuiti sugli stessi comuni, mantenendone posizione e peso.

I parametri utilizzati per la generazione della traiettoria tramite le due configurazioni sono:

- Modello Terra piatta
- Area di ricerca di 60 km x 60 km
- Ricerca monodirezionale
- Ricerca discreta con griglia da 240 x 240
- Lunghezza del passo di 3,5 km
- Raggio di curvatura minimo di 2 miglia nautiche
- Coefficiente bias di distanza nullo
- 5 nuovi nodi per espansione

La [Figura 6.3](#) presenta i grafi che mostrano i percorsi generati e i nodi visitati dalle due configurazioni.

Risultano evidenti le forti differenze nei due percorsi generati dalle due configurazioni.

In questo caso, la Configurazione 1 tende a visitare nuovi nodi con un ordine molto influenzato dalla distanza percorsa. Il percorso generato non è infatti dissimile dal percorso più breve dal punto iniziale all'obiettivo, in termini di distanza.

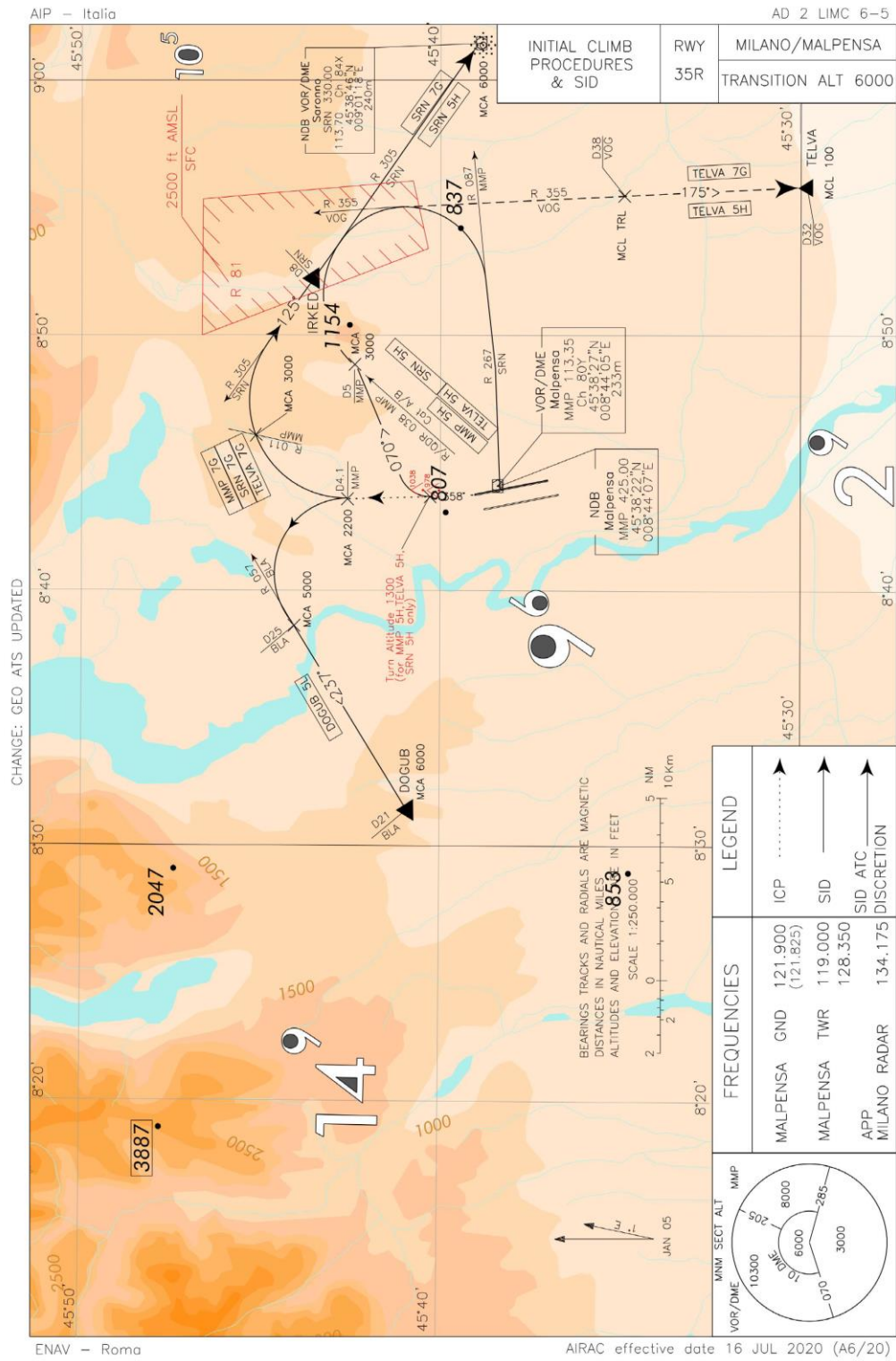


Figura 6.1: Carta Aeronautica in cui la SID DOGUB5L è visibile

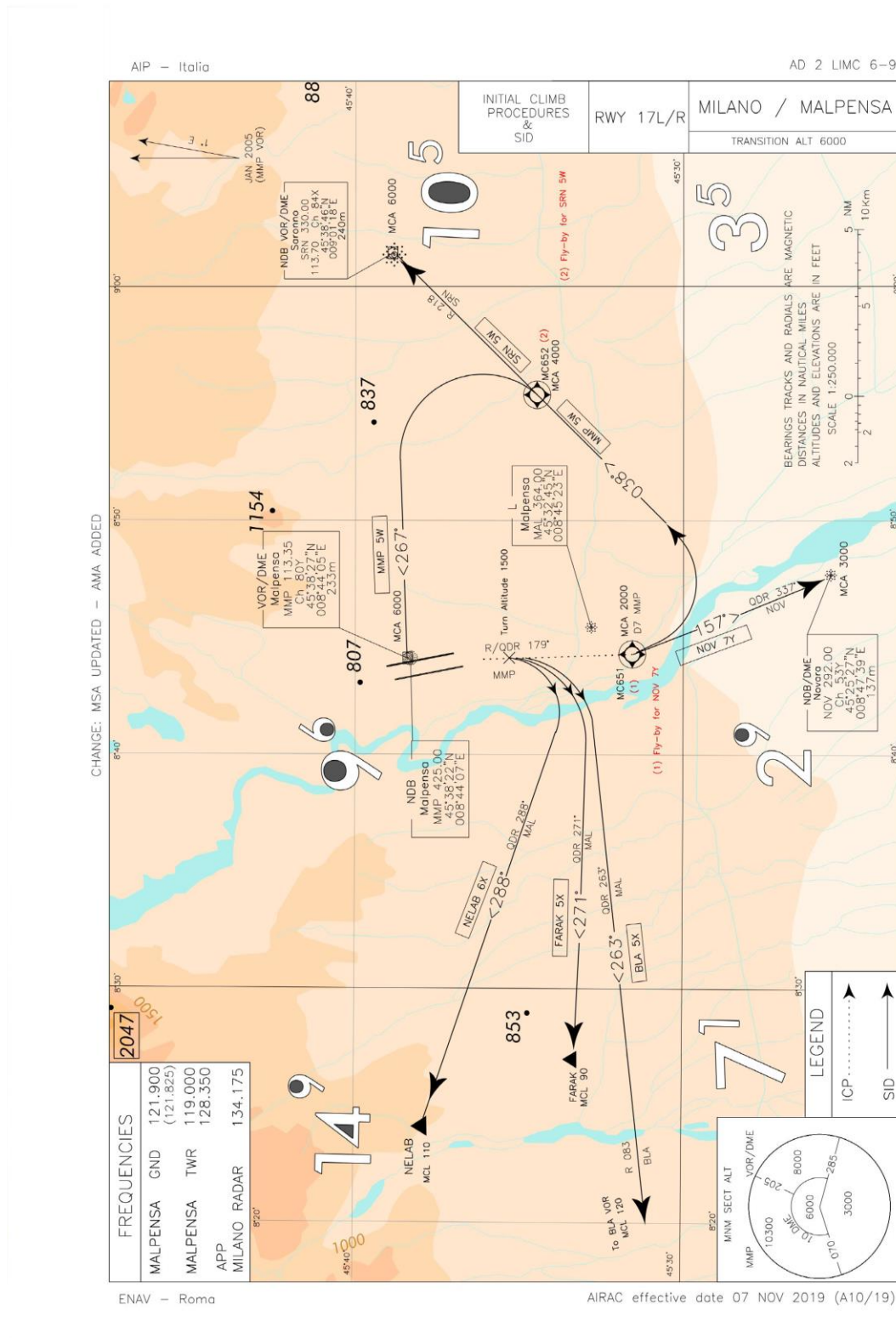


Figura 6.2: Carta Aeronautica in cui la SID SRN5W è visibile

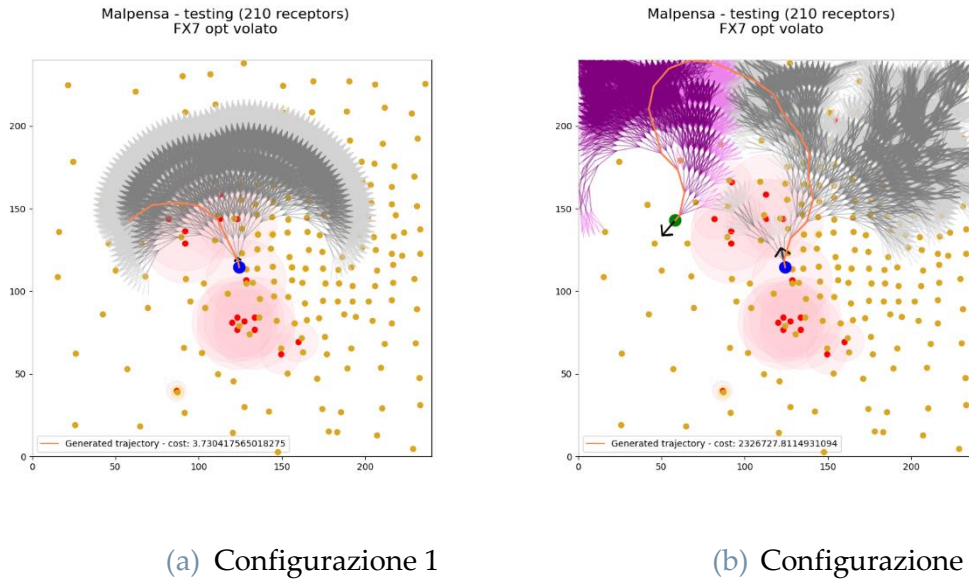


Figura 6.3: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b)

La configurazione 2, al contrario, spinge la ricerca lontano dai nodi centrali in quanto l'euristica utilizzata e la funzione di rumore associano dei valori troppo vantaggiosi ai nodi che si allontanano dai recettori, facendo incontrare i due grafi nei pressi della fine dell'area di ricerca e risultando in un percorso troppo lungo per essere percorso in un caso reale.

6.3.1.2. Analisi delle prestazioni

La Tabella 6.2: Tempo impiegato e nodi visitati dalle due configurazioni per raggiungere la soluzione mostra i dati relativi al tempo impiegato per la generazione del percorso e a riguardo del numero di nodi visitati per arrivare ad una soluzione

Tabella 6.2: Tempo impiegato e nodi visitati dalle due configurazioni per raggiungere la soluzione

	Tempo [s]	Nodi	Rateo [iter/s]
Config 1	35,65	6630	185,97
Config 2	498,26	10980	22,04

La configurazione 1 riesce ad essere più efficiente sia per numero di nodi visitati, come già evidenziato, che per numero di nodi visitati al secondo. Ciò risulta in un guadagno temporale del 1398% in questa configurazione.

6.3.1.3. Confronto con SID DOGUB5L

La Figura 6.4 presenta una sovrapposizione tra la traiettoria generata dall' algoritmo utilizzando la Configurazione 1 e la rotta seguita dal volo FX7/N865FD. Figura 6.4: Confronto tra il percorso generato dalla Configurazione 1 (in arancione) e la traiettoria seguita dal volo

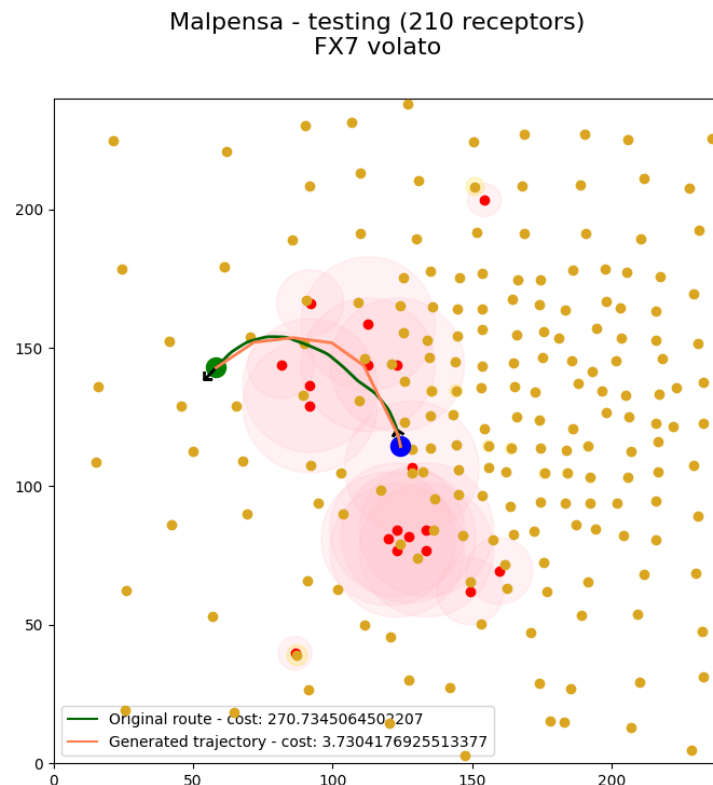


Figura 6.4: Confronto tra il percorso generato dalla Configurazione 1 (in arancione) e la traiettoria seguita dal volo FX7/N865FD (in verde)

Risulta evidente una similarità tra i due percorsi. In particolare, entrambi tendono a diminuire la distanza totale percorsa sfruttando il corridoio formato dai comuni a nord del sito dell'aeroporto.

La presenta un confronto tra i costi dei percorsi generati tramite le due configurazioni e il costo della traiettoria realmente volata.

La traiettoria prodotta dalla nuova configurazione risulta, nel costo, molto più contenuta, riuscendo ad impattare del 98,6% in meno rispetto alla traiettoria volata. Il vantaggio rispetto alla traiettoria generata con la precedente implementazione è

evidente, riuscendo essa a generare un percorso meno rumoroso del 77,6% di quello volato.

Tabella 6.3: Confronto tra i costi dei percorsi generate con le due configurazioni e il percorso originale

	Rumore generato	Differenza con rotta originale
Configurazione 1	3,734	-98,6%
Configurazione 2	60,4957	-77,6%

6.3.2. Partenza SID SRN5W

Il secondo caso di studio si concentra su un decollo che segue la traiettoria individuata dal SID SRN5W, visibile in [Figura 6.2](#).

Il velivolo decolla, in questo caso, dalla pista 17R procedendo verso sud e raggiungendo il punto di uscita situato a est della pista.

Al fine di rendere possibile un paragone sono stati usati dati del volo KZ49/NCA049 per Tokyo (NRT), con partenza da Malpensa (MXP) alle 21.28 del 21 Marzo 2021.

La tabella presenta le coordinate iniziali e finali, con i rispettivi azimuth, del punto di partenza e di arrivo della rotta.

Tabella 6.4: Configurazione nodo iniziale e finale della rotta KZ49/NCA049

	Posizione iniziale	Posizione finale
Latitudine	45.644840 °	45.647627°
Longitudine	8.718823°	9.048991°
Azimuth	168°	60°

6.3.2.1. Analisi dei percorsi

Anche in questo caso il peso e la posizione dei recettori sensibili sono rimasti invariati per permettere un confronto prestazionale rispetto alla nuova configurazione.

I parametri utilizzati per la generazione della traiettoria tramite le due configurazioni sono:

- Modello Terra piatta
- Area di ricerca di 60 km x 60 km
- Ricerca monodirezionale
- Ricerca discreta con griglia da 240 x 240
- Lunghezza del passo di 3,5 km
- Raggio di curvatura minimo di 2 miglia nautiche
- Coefficiente bias di distanza nullo
- 5 nuovi nodi per espansione

In Figura 6.5 sono presentati i risultati dell'esecuzione delle due configurazioni senza limitazioni sull'area di ricerca.

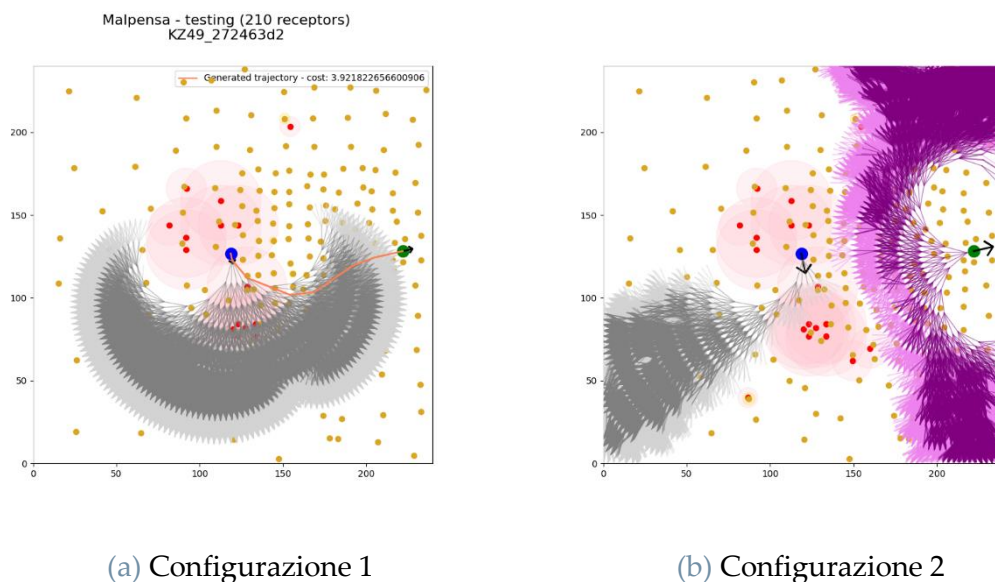


Figura 6.5: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b) senza limiti nell'area di ricerca.

La Configurazione 2 non giunge a terminazione.

In seguito ad esecuzione della durata di 50 minuti, l'esecuzione della Configurazione 2 è stata bloccata prima di arrivare a terminazione.

Anche in questo caso il comportamento, per quanto riguarda i nodi visitati, delle due configurazioni è simile al caso di studio precedente.

La Configurazione 1 tende a visitare nuovi nodi in un ordine uniforme, pur protraendo maggiormente, come atteso, le zone che presentano meno recettori sensibili. Diversamente, la Configurazione 2 tende ad estendere il percorso prediligendo i

rami che si allontanano dai recettori, invece che preferire i rami che hanno più probabilità di portare a terminazione l'esecuzione.

Tale comportamento risulta evidente nella [Figura 6.5b](#), in cui i due alberi di ricerca faticano ad incontrarsi in quanto le direzioni in cui si espandono non permettono di incrociarsi.

In particolare, l'albero che parte dal nodo iniziale (in grigio) tende a virare quasi immediatamente verso ovest al fine di allontanarsi dall'agglomerato centrale di recettori sensibili.

Per lo stesso motivo, l'albero opposto (in viola) si espande contemporaneamente verso nord e verso sud.

Al fine di confrontare i percorsi generati, le due configurazioni sono state eseguite riducendo l'area di ricerca con le seguenti impostazioni: 2km nord, 30km sud, 10km ovest, 0m est.

L'area è dimensionata per permettere alla Configurazione 2 di continuare ad espandere i nodi nelle direzioni in cui sono stati espansi nella precedente esecuzione, tagliando però le zone più limitrofe, rendendo più semplice arrivare a terminazione.

In [Figura 6.6](#) sono presentati i risultati dell'esecuzione delle due configurazioni limitando l'area di ricerca.

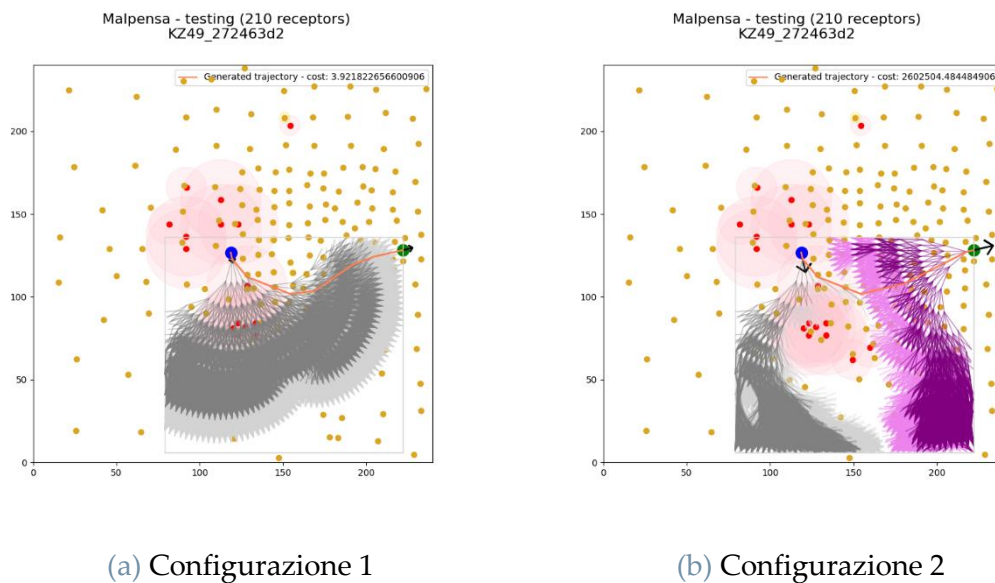


Figura 6.6: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b) limitando l'essenzione dell'area di ricerca.

In questo caso, il percorso generato dalle due configurazioni risulta simile. In entrambi i casi, la traiettoria prevede di virare verso est immediatamente dopo aver lasciato la pista di decollo dirigendosi verso il punto di uscita.

Risulta comunque problematico per la Configurazione 2 concentrare la ricerca nell'area in cui si trova, successivamente, il percorso più breve.

6.3.2.2. Analisi delle prestazioni

La Tabella 6.5: Tempo impiegato e nodi visitati dalle due configurazioni senza limiti di area mostra i dati relativi al tempo impiegato e il numero di nodi visitati nel caso della ricerca senza limiti di area.

Tabella 6.5: Tempo impiegato e nodi visitati dalle due configurazioni senza limiti di area

	Tempo [s]	Nodi	Rateo [iter/s]
Config 1	137,64	24808	137,64
Config 2	3000+	39550+	13,18

In questo caso, solo la Configurazione 1 riesce ad arrivare a terminazione in tempi accettabili.

È importante evidenziare come non solo il rateo di visita di nuovi nodi sia molto più alto ma il numero di nodi visitati necessario alla terminazione dell'algoritmo è nettamente inferiore nella Configurazione 1.

La Tabella 6.6 mostra i dati relativi al tempo impiegato e il numero di nodi visitati nel caso della ricerca senza limiti di area.

Tabella 6.6: Tempo impiegato e nodi visitati dalle due configurazioni con limiti di area

	Tempo [s]	Nodi	Rateo [iter/s]
Config 1	73,70	15360	208,42
Config 2	409,83	12155	29,66

Anche se il numero di nodi visitati è più alto, in questo caso, nella Configurazione 1, quest'ultima si conferma più stabile nella generazione di un percorso in casi complicati.

Ciò è possibile grazie alla distribuzione dei nodi più uniforme e al rateo di visita che risulta di un ordine di grandezza più rapido rispetto alla Configurazione 2. Per questi motivi, il guadagno temporale in questo caso è del 556%.

6.3.2.3. Confronto con SID SRN5W

La presenta una sovrapposizione tra la traiettoria generata dall'algoritmo utilizzando la Configurazione 1 e la rotta seguita dal volo KZ49/NCA049.

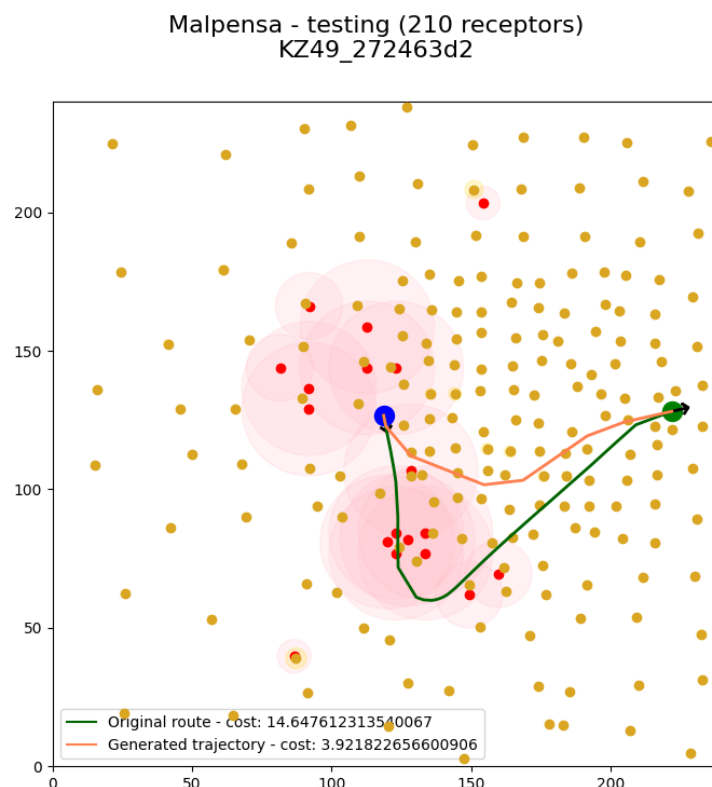


Figura 6.7: Confronto tra il percorso generato dalla Configurazione 1 (in arancione) e la traiettoria seguita dal volo KZ49/NCA049

È inoltre evidente come la rotta volata passi proprio sopra il gruppo di recettori sensibili a sud della pista, risultando in un costo molto più elevato rispetto alla rotta generata.

È inoltre evidente come la rotta generata risulta molto più breve della rotta volata dal KZ49/NCA049, il che risulta positivo per quanto riguarda il carburante necessario per percorrerla, la cui diminuzione ha effetti sia ambientali che economici.

La Tabella 6.7 presenta un confronto in termini di rumore tra i percorsi generati con le due configurazioni e il percorso realmente volato.

I costi dei due percorsi generati sono pressoché uguali, aggirandosi entrambi su una riduzione del rumore totale del 73%.

In questo caso il vantaggio offerto dalla nuova implementazione è esclusivamente

prestazionale.

È importante notare, però, che il percorso così generato dalla configurazione 2 è forzatamente ottenuto limitando l'area di ricerca nei dintorni della traiettoria ottenuta con la nuova implementazione non riuscendo, la configurazione 2, a terminare altrimenti.

Tabella 6.7: Confronto tra i costi dei percorsi generate con le due configurazioni e il percorso originale

	Rumore generato	Differenza con rotta originale
Configurazione 1	3,921	-73,2%
Configurazione 2	3,857	-73,6%

7 Conclusioni

7.1. Risultato delle analisi

Il lavoro svolto in questa tesi è indirizzato al miglioramento delle traiettorie generate dalla precedente implementazione in termini di rumore percepito sulla popolazione.

L'approccio utilizzato a questo fine ha previsto sia un'espansione concettuale del modello in uso, che delle modifiche atte al miglioramento delle prestazioni temporali e in termini di nodi totali visitati dall'algoritmo.

L'incremento delle performance permette di utilizzare una lunghezza del passo minore e di incrementare i nodi considerati per ogni espansione, ottenendo un aumento nell'accuratezza del percorso prodotto nella fase generativa.

7.1.1. Modifiche al modello

L'espansione del modello include l'introduzione della terza dimensione tramite l'applicazione dell'aumento di quota costante, il che ha permesso di generare dei percorsi che minimizzano con maggiore verosimiglianza il rumore percepito a livello del suolo lungo tutta la lunghezza della traiettoria.

Dal punto di vista prestazionale, l'introduzione risulta in una diminuzione di nodi visitati senza inficiare sul rateo di visita di nuovi nodi.

L'utilizzo dei percorsi di Dubins per calcolare la distanza minima applicati alla ricerca di una sovrapposizione tra due nodi ha permesso il rilevamento di traiettorie percorribili anche nella parte finale prossima al nodo obiettivo, scartando con sicurezza i percorsi i cui nodi finali non possono raggiungere l'obiettivo per via del vincolo imposto dal raggio massimo di curvatura.

L'applicazione dei percorsi al calcolo dell'euristica, pur diminuendo in media nodi visitati e il tempo di esecuzione, non ha generato risultati tali da giustificare il mantenimento dell'euristica nell'algoritmo.

Il modello tridimensionale introdotto, in aggiunta al modello direzionale di rumore, è risultato in una modifica sostanziale del percorso generato, conseguito da un aumento del realismo delle scelte effettuate dall'algoritmo.

La nuova funzione di rumore ha contribuito a diminuire il tempo impiegato tramite il taglio di nodi visitati causato da quest'ultima.

7.1.2. Aumento prestazionale

La modifica dell'euristica tramite l'applicazione dei percorsi di Dubins non ha prodotto un miglioramento dei tempi tale da giustificare il costo implementativo di A^* rispetto all'algoritmo di Dijkstra.

La nuova euristica basata su un coefficiente e la distanza dal nodo obiettivo, invece, pur preservando il rateo di nodi visitati al secondo, non ha avuto un impatto tale da giustificare l'implementazione, a causa del valore troppo basso ad essa associato.

L'utilizzo dell'algoritmo di Dijkstra è risultato più efficiente dal punto di vista prestazionale. Esso permette di avere una soluzione in tempi più brevi sviluppando l'albero di ricerca in modo più distribuito, risultando più consono al contesto di generazione dinamica del grafo.

L'introduzione del filtro passa-alto ha permesso di ridurre l'impatto che la quantità di recettori ha sul processo di campionamento, diminuendo il tempo totale impiegato dall'algoritmo.

La ricerca bidirezionale è stata eliminata in quanto non compatibile con l'aumento di altitudine costante, la quale si è scelta di implementare in quanto risultata, però, fondamentale per aumentare la correttezza delle traiettorie generate.

L'utilizzo della ricerca monodirezionale ha aumentato la complessità teorica dell'algoritmo che però, nel complesso, ha visto diminuire i tempi di esecuzione.

7.1.3. Considerazioni generali

La nuova configurazione ottenuta al netto delle modifiche implementate offre prestazioni migliori nella generazione di una traiettoria anche non imponendo una riduzione dell'area di ricerca, a differenza della precedente configurazione dell'algoritmo.

Ciò apporta un livello di generalizzazione sfruttabile per avere buone prestazioni anche nelle configurazioni più complesse.

L'algoritmo presenta ancora delle problematiche che interessano l'espansione del grafo verso zone che si allontanano dall'obiettivo, pur in modo radicalmente minore rispetto alla precedente implementazione.

In tali zone, la generazione dei nodi assume un carattere ridondante, andando a coprire più volte la stessa area.

La riduzione di quest'ultimi tramite l'implementazione di una euristica più

adeguata al caso sarebbe di primaria importanza per lo sviluppo dell'algoritmo generativo in lavori futuri.

Nei casi di studio considerati, le traiettorie generate dalla nuova configurazione risultano molto simili alle rotte realmente utilizzata nel contesto di Malpensa, il che può essere considerato un segno positivo per quanto riguarda la validazione dell'efficacia dell'algoritmo nel creare traiettorie ottimizzate da utilizzare anche in contesti differenti.

In generale, però, lo sviluppo attuale non permette ancora l'utilizzo dell'algoritmo in contesti reali.

Questa tesi è intesa come un punto di partenza per uno sviluppo ulteriore del progetto.

Il prossimo capitolo presenterà possibili idee da cui partire per sviluppi futuri, a partire dalle analisi svolte.

La tesi fa parte di un progetto che TXT e-solutions sta portando avanti per migliorare la qualità della vita delle comunità interessate dal fenomeno dell'inquinamento acustico generato dalla presenza di aeroporti.

Grazie all'esperienza nell'industria aeronautica, TXT sarà in grado di continuare il progetto fino al punto in cui sarà pronto per essere proposto come strumento alternativo alle autorità competenti.

7.2. Sviluppi futuri

7.2.1. Ottimizzazioni computazionali

Ad oggi, l'implementazione utilizzata offre delle prestazioni non ancora consone ad un utilizzo in casi reali.

I nodi visitati sono eccessivi in numero e concentrati, in gran parte, nelle aree limitrofe dell'area di ricerca.

Di seguito sono presentate delle possibili idee che porterebbero al miglioramento dell'attuale implementazione, da sviluppare in progetti.

7.2.1.1. Modello tridimensionale con ricerca bidirezionale

La ricerca bidirezionale offriva una netta diminuzione del numero di nodi visitati ma si è rivelata incompatibile con il modello tridimensionale.

La reintroduzione della ricerca bidirezionale apporterebbe un sicuro miglioramento prestazionale nel caso in cui si riuscissero a risolvere i problemi legati alla precedente implementazione, i quali:

- Incertezza sul punto di partenza dell'albero di ricerca opposto, causata dal fatto che l'altezza del nodo obiettivo non è definita a priori.
Una futura versione della ricerca bidirezionale dovrebbe tenere in considerazione diverse altezze finali oppure non tenere in considerazione tale parametro.
- Dipendenza del percorso trovato dall'area di ricerca utilizzata a causa delle approssimazioni utilizzate.

7.2.1.2. Euristiche

La reintroduzione di un'euristica consona al caso d'esame ridurrebbe il numero di nodi considerati.

Una nuova euristica dovrebbe essere allo stesso tempo:

- Rilevante dal punto di vista del valore assegnato ad essa, rapportato al costo del percorso già esplorato. Ciò è necessario al fine di evitare che il suo apporto nella definizione della coda di nodi da visitare nulla o irrilevante.
- Poco pesante dal punto di vista dei calcoli necessari al suo ottenimento, al fine di non inficiare eccessivamente sul rateo di visita dei nuovi nodi.
- In linea con la condizione di terminazione dell'algoritmo, cioè il raggiungimento del nodo obiettivo.

Un problema legato alla generazione di nodi in aree non utili anche in presenza dell'euristica utilizzata fin ora era il fatto che il contributo dato dalla

stessa prediligeva i nodi che si allontanavano dai recettori, invece che i nodi che si avvicinavano all'obiettivo.

7.2.2. Espansione modello

7.2.2.1. Compromesso tra ottimizzazione e lunghezza

L'aumento del consumo di carburante associato alla generazione di una traiettoria più lunga delle varianti realmente in uso ha effetti sia ambientali che economici. Un'implementazione dell'algoritmo utilizzabile in un contesto reale dovrebbe offrire la possibilità di tenere in conto della lunghezza della tratta imponendo un compromesso tra quest'ultima e il rumore generato.

Attualmente non è possibile imporre una limitazione legata alla lunghezza per via dei problemi imp

7.2.2.2. Considerazione di norme di volo strumentale

Le rotte generate, allo stato attuale, sono basate sulle possibilità di volo libero di un velivolo e richiedono, per essere volate, una strumentazione di bordo che permetta di seguire dei punti espressi in coordinate geografiche.

La strumentazione standard richiesta dalla maggioranza degli aeroporti è, però, basata su radiofrequenze.

Queste permettono di seguire rotte formate da linee rette o da archi di circonferenze aventi lo stesso centro, situato in un *Turning Point*.

Al fine di permettere di seguire, con una certa precisione, le rotte generate dall'algoritmo, è necessario integrare nell'implementazione dello stesso un modo per tenere conto delle diverse possibilità tecniche di navigazione strumentale.

Una versione dell'algoritmo che tiene conto di ciò dovrebbe integrare nel modello i diversi tipi di radioassistenze, considerandone posizione geografica e possibilità offerte in termini di percorso, limitando i tracciati percorribili in base a queste ultime.

7.2.2.3. Integrazione con modelli più realistici della terra

Le innovazioni apportate da questa tesi sono state implementate utilizzando il modello della Terra piatta al fine di permetterne lo studio dei vantaggi apportati. Le operazioni riguardanti le modifiche implementate dovrebbero comprendere il modello sferico e il modello geoidale della Terra, per offrire una maggiore precisione delle soluzioni ad ogni latitudine e longitudine.

La maggiore problematica è rappresentata dall'integrazione dei percorsi di Dubins con tali modelli, nonché dall'utilizzo delle formule di Haversine e di Vincenty in un contesto che tiene conto dell'altitudine.

Bibliografia

- BBC News. (2009, 09 10). *Road noise link to blood pressure*. Tratto da BBC News: <http://news.bbc.co.uk/2/hi/health/8247217.stm>
- Dubins, L. E. (1957). On Curves of Minimal Length with a Constraint on Average Curvature, and with. *American Journal of mathematics*, 497-516.
- EU. (2014, april 16). Regulation 598/2014 of the European Parliament and the Council of 16 April 2014 on the establishment of rules and procedures with regard to the introduction of noise related operation restriction at Union airport within Balanced Approach. *Official Journal of the European Union*, 65-79.
- FAA. (2022, 03 01). *Community responce to Noise*. Tratto da faa.gov: https://www.faa.gov/regulations_policies/policy_guidance/noise/community/
- FAA. (2022, 03 01). *Fundamentals of Noise and Sound*. Tratto da faa.gov: https://www.faa.gov/regulations_policies/policy_guidance/noise/basics/
- Gullo, M. (2021). *Air Trajectory Optimization for Noise Reduction*. Milano: Politecnico di Milano.
- Hameed, I. A. (2017, July). Coverage Path Planning Software for Autonomous Robotic Lawn Mower using Dubins' Curve. *IEEE International Conference on Real-time Computing and Robotics (RCAR)* (p. 517-522). IEEE.
- Kerns, E. M. (2018). Cardiovascular conditions, hearing difficulty, and occupational noise exposure within US industries and occupations. *American journal of industrial medicine*, 477-491.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge university press.
- Münzel T, S. F. (2018). Environmental noise and the cardiovascular system. *Journal of the American College of Cardiology*, 71(6), 688-697.
- N. van Oosten, I. A. (2009). *Background noise levels and noise levels from en-route aircraft*. EASA. Tratto da <https://www.easa.europa.eu/downloads/3854/en>

- Niu, L. & Zhuo, G. (2008). An improved real 3D A* algorithm for difficult path finding situation. *Proceeding of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, (p. 37).
- Python. (s.d.). *Minimize method SLSQP*. Tratto da docs.scipy.org: <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html>
- Rosen, S. &. (1965). Hearing Loss and Coronary Heart Disease. *Archives of Otolaryngology*, 82(3), 236-243.
- Schiavina, M. F. (2019). *GHS-POP R2019A - GHS population grid multitemporal (1975-1990-2000-2015)*. European Commission, Joint Research Centre (JRC). doi:10.2905/0C6B9751-A71F-4062-830B-43C9F432370F
- SEA s.p.a. (2013, Luglio 18). *Piano d'Azione ai sensi del D.Lgs. 194/05*. Tratto da seamilano.eu: https://seamilano.eu/sites/default/files/imce/piano_dazione_aeroporto_milano_malpensa_sea_s.p.a._18_luglio_2013.pdf
- Shkel, A. M. (1996, April). On Calculation of Optimal Paths with Constrained Curves: The Case of Long Paths. *Proceedings of IEEE International Conference on Robotics and Automation* (p. 3578-3583, Vol. 4). IEEE.
- Shkel, A. M. (2001). Classification of the Dubins set. *Robotics and Autonomous Systems*, 34(4), 179-202.
- Smart, R. R. (2008). *Evolutionary Control of Autonomous Underwater Vehicles*. RMIT University.
- Wlaker, A. (2011). *Hard Real-Time Motion Planning for Autonomous Vehicles*". Swiburne University.
- Zambrini, S. (2012). *La misura della qualità di vita nei luoghi e nell'ambiente. Metodologia e criteri di valutazione*. Santarcangelo di Romagna: Maggioli Editore.
- Zepeda, P. (2022). *Aircraft sound modeling through simulation for route optimization*. Milano: Politecnico di Milano.

Lista delle Figure

Figura 2.1: Scala di riferimento decibel	8
Figura 2.2: Scala di riferimento per valori DNL, espressa in dB.....	10
Figura 3.1: punti chiave dell'Approccio Bilanciato.....	11
Figura 4.1: Differenza nella generazione dei nodi utilizzando A* e l'algoritmo di Dijkstra nella configurazione 4	24
Figura 4.2: Un confronto tra gli alberi di ricerca generati da A* e dall'algoritmo di Dijkstra	27
Figura 4.3: Alberi generati con distanza cartesiana come distanza minima	29
Figura 4.4: Alberi generati con distanza di Dubins come distanza minima	30
Figura 4.5: Alberi generati con algoritmo di Dijkstra.....	31
Figura 4.6: Percorso generato dall'algoritmo in cui è presente una terminazione errata causata dalla funzione di controllo di giunzione	37
Figura 5.1: I sei possibili percorsi di Dubins da un punto iniziale qs a un punto finale qf	40
Figura 5.2: Percorsi generati sulla stessa configurazione ma ad altitudini differenti	45
Figura 5.3: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso a.....	51
Figura 5.4: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso b	51
Figura 5.5: Visualizzazione grafica dell'area d'effetto dell'intensità sonora con una soglia di 50dB – caso c.....	52
Figura 5.6: Percorsi generate sulla stessa configurazione utilizzando un numero diverso di nodi per espansione.....	61
Figura 6.1: Carta Aeronautica in cui la SID DOGUB5L è visibile.....	68
Figura 6.2: Carta Aeronautica in cui la SID SRN5W è visibile.....	69

Figura 6.3: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b)	70
Figura 6.4: Confronto tra il percorso generato dalla Configurazione 1 (in arancione) e la traiettoria seguita dal volo FX7/N865FD (in verde)	71
Figura 6.5: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b) senza limiti nell'area di ricerca. La Configurazione 2 non giunge a terminazione.	73
Figura 6.6: Percorso generato e nodi visitati dalle esecuzioni utilizzando la configurazione 1 (a) e la configurazione 2 (b) limitando l'esenzione dell'area di ricerca.	74
Figura 6.7: Confronto tra il percorso generato dalla Configurazione 1 (in arancione) e la traiettoria seguita dal volo KZ49/NCA049	76

Lista delle Tabelle

Tabella 4.1: Configurazioni utilizzate per i test.....	23
Tabella 4.2: Nodi visitati nel caso di A* e dell'algoritmo di Dijkstra su diverse configurazioni	23
Tabella 4.3: Confronto dei tempi necessari a raggiungere una soluzione nel caso di A* e dell'algoritmo di Dijkstra.....	25
Tabella 5.1: Distanza alla quale si raggiunge una certa intensità sonora con un motore turbojet.....	49
Tabella 5.2: Distanza alla quale si raggiunge una certa intensità sonora con un motore turbofan	50
Tabella 5.3: Numero di nodi visitati da A* utilizzando la distanza cartesiana e di Dubins per la generazione della configurazione fittizia.....	56
Tabella 5.4: Confronto dei tempi necessari a raggiungere una soluzione utilizzando la distanza cartesiana e di Dubins per la generazione della configurazione fittizia	57
Tabella 6.1: Configurazione nodo iniziale e finale della rotta FX7/N8865FD	67
Tabella 6.2: Tempo impiegato e nodi visitati dalle due configurazioni per raggiungere la soluzione	70
Tabella 6.3: Confronto tra i costi dei percorsi generate con le due configurazioni e il percorso originale	72
Tabella 6.4: Configurazione nodo iniziale e finale della rotta KZ49/NCA049.....	72
Tabella 6.5: Tempo impiegato e nodi visitati dalle due configurazioni senza limiti di area.....	75
Tabella 6.6: Tempo impiegato e nodi visitati dalle due configurazioni con limiti di area.....	75
Tabella 6.7: Confronto tra i costi dei percorsi generate con le due configurazioni e il percorso originale	77

