



UNIVERSITÀ DI PERUGIA  
Dipartimento di Matematica e Informatica



### TESI MAGISTRALE IN INFORMATICA

## L'instradamento degli octocopter alla luce delle condizioni del vento

*Relatore*

**Prof.ssa Maria Cristina Pinotti**

*Candidato*

**Lorenzo Palazzetti**

---

Anno Accademico 2019-2020

*“Nel coltivare sé stessi, non esiste la parola «fine».  
Chi si ritiene completo, in realtà, ha voltato le spalle alla Via.”*

*- Yamamoto Tsunetomo.*

# Capitolo 1

## Introduzione

Negli ultimi anni, i droni, detti anche UAV (Unmanned Aerial Vehicles), sono stati ampiamente adoperati in applicazioni civili come agricoltura di precisione, monitoraggio, localizzazione di oggetti o persone, e consegna delle merci. In quest'ultimo caso grandi colossi aziendali come ad esempio, Amazon, Google, UPS e DHL stanno già testando sistemi di consegna che integrano al loro interno i droni, in particolare focalizzandosi sulla logistica denominata "dell'ultimo miglio", principalmente concentrata su piccoli pacchetti da consegnare. Ci sono numerosi vantaggi nell'utilizzare i droni nel contesto delle consegne, sia dal punto di vista dei benefici economici che in termini di capacità di lavoro in situazioni di tempo critico o luoghi difficili da raggiungere. Considerando queste potenzialità, società di trasporto possono ampliare ulteriormente la propria attività affidandosi a droni che consegnano merci direttamente alle sedi dei clienti in città o aree di medie e grandi dimensioni.

Nello scenario che si intende studiare si assume di avere un deposito (magazzino) e una flotta di droni per servire i clienti che sono sparsi nell'area di consegna descritta dal suolo della Corsica. Poiché il limite di carico utile costringe il drone a tornare al deposito dopo ogni consegna, il problema del commesso viaggiatore (Traveling Salesman Problem, TSP) non è in grado di modellare correttamente il sistema di consegna definito; quindi è possibile servire ogni cliente solo separatamente. Il drone che intende intraprendere una consegna, pianificherà il tragitto di minimo costo energetico sulla base della mappa di instradamento costruita per i suoi spostamenti e delle condizioni meteorologiche attuali (in particolare, il vento) che pervadono l'ambiente, al fine di garantire un maggiore risparmio energetico. In volo infatti, il consumo di

energia è pesantemente influenzato dalla direzione e dalla velocità dei venti presenti, che a loro volta sono influenzati dagli edifici e dagli ostacoli in generale dislocati sul territorio. Il drone, sotto queste condizioni, preferirà volare con il vento a favore perché richiede meno energia rispetto che volare con il vento contrario. Altra voce di costo è il peso del carico che influisce sull'autonomia della batteria del drone durante il suo volo.

L'obiettivo che questo lavoro si prefigge è quello di investigare come reali condizioni del vento, registrati da diverse stazioni meteo della Corsica, possano in qualche modo impattare in termini energetici in un contesto di consegna di merci mediante l'utilizzo di un drone. Si concentrerà l'attenzione principalmente sul problema di adempiere ad una consegna trovando un ciclo per il drone che sia fattibile, cioè, che possa essere completata con l'autonomia energetica disponibile del drone. Il consumo energetico del drone dipenderà dai parametri statici e dinamici dello scenario e sulla base di questi il drone potrà decidere, durante la missione, di prendere percorsi diversi da quelli originariamente previsti per sfruttare i cambiamenti del vento.

Il resto della tesi è così articolata: nel Capitolo 2 verrà data una descrizione generica dello stato dell'arte riguardante i modelli di droni attualmente esistenti sul mercato e le loro applicazioni; nel Capitolo 3 si introdurrà al Mission Feasibility Problem che rappresenta la modellazione naturale del caso reale che si vuole studiare e verranno descritti i concetti teorici su cui poggerà l'implementazione pratica costruita; nel Capitolo 4 verranno descritte le tre procedure su cui si fondono le strategie di spostamento del drone al fine di effettuare le consegne e quindi su cui si baseranno le simulazioni; nel Capitolo 5 sarà descritta innanzitutto l'architettura e la configurazione che sta alla base delle simulazioni e infine sarà svolta un'attenta analisi dei risultati ottenuti; nel Capitolo 6 sono prospettate alcune conclusioni e possibili sviluppi futuri.

# Capitolo 2

## L’uso dei droni nella logistica

Gli *unmanned aerial vehicles* (UAVs), ossia, veicoli a pilotaggio remoto comunemente detti *droni*, sono una tecnologia molto promettente che potrebbe apportare significative ripercussioni all’interno delle dinamiche di mercato a livello mondiale. Il principale vantaggio che deriva dal loro utilizzo in ambito civile è il sostanziale risparmio economico in termini di consumi energetici e di manodopera. All’interno di questo capitolo verrà fornita una descrizione delle differenti tipologie di dispositivi attualmente in commercio, andando a delineare dettagliatamente, in relazione alle loro caratteristiche, i diversi ambiti di applicazione. La trattazione si concluderà con la definizione delle principali formulazioni di problema attualmente conosciute.

### 2.1 Classificazione dei droni

Nel corso del tempo è stato riscontrato un sempre più crescente incremento delle applicazioni dei droni sia in ambito civile che in ambito militare. Questo aumento di soluzioni, che integrano all’interno dei loro processi i droni, è andato di pari passo con i progressi tecnologici ottenuti nel campo delle batterie e delle telecomunicazioni. In questo scenario di prosperità tecnologica sono stati sviluppati una vasta gamma di modelli e configurazioni, che in relazione ai vincoli e alle restrizioni imposte dall’ambiente operativo in cui si sta lavorando, potevano adattarsi in modo più aderente alle necessità e garantire migliori performance.

Innanzitutto definiamo un UAV come un apparecchiatura in grado di volare e caratterizzata dall’assenza di un pilota a bordo. Il suo volo è controllato dal computer

a bordo del mezzo oppure tramite il controllo remoto di un navigatore o pilota, sul terreno o in altre posizioni. Questi dispositivi a seconda della missione a cui devono adempiere hanno una struttura e una configurazione che varia rispetto a questi parametri:

- dimensione dell'aeromobile;
- distanza massima di viaggio;
- capacità di curvatura e atterraggio;
- capacità di trasporto.

Tuttavia seppure queste caratteristiche possano essere discriminanti per delineare la fisionomia dei diversi droni attualmente in commercio, esistono ulteriori modi per catalogarli.

Arjomandi et al. [1] ha classificato i droni in base a peso, autonomia, resistenza, carico alare, altitudine massima e tipo di motore. All'interno dell'articolo vengono dunque classificati i droni come super-pesanti con peso superiore a 2000 kg, pesanti con peso compreso tra 200 kg e 2000 kg, medi con peso compreso tra 50 kg e 200 kg, leggeri/mini con peso compreso tra 5 kg e 50 kg, e infine micro droni con peso inferiori a 5 kg.

Gupta et al. [12] classifica i droni come HALE (High Altitude Long Endurance), MALE (Medium Altitude Long Endurance), TUAV (medium range o tactical UAV), MUAV o Mini UAV, MAV, and NAV (Nano Air Vehicles) sulla base del loro equipaggiamento, della massima altitudine e distanza raggiungibile.

Tuttavia non esiste una classificazione standard di questi dispositivi ed anzi esistono una miriade di modi per classificarli, in quanto altrettanti modelli con le più svariate proprietà sono stati costruiti negli ultimi anni. Quindi seppure il panorama sia molto variegato è possibile separare in due grandi famiglie di appartenenza i droni:

- **Fixed-wing:** questa tipologia di drone si riferisce ad aerei senza equipaggio (con ali) riconducibili per forma a veri e propri aerei (Figura 2.1a). Generalmente richiedono un supporto per decollare ed atterrare, tuttavia una volta in volo riescono a percorrere tratti di cielo molto ampi, riuscendo a volare anche ad alte velocità di crociera.

- **Rotorcraft:** chiamati anche UAV rotorcraft o UAV di decollo e atterraggio verticale (VTOL), hanno i vantaggi di essere particolarmente stabili in volo, tanto da essere in grado di librare in aria e di possedere una elevata manovrabilità. Queste capacità sono funzionali per molteplici tipi di missione, specialmente in applicazioni civili, come ad esempio le consegne. Un rotor-UAV (Figura 2.1b) può avere diverse configurazioni, con rotori principali e di coda (convenzionale elicotteri), rotori coassiali, rotori tandem, multi-rotori, ecc. Nello specifico, il Parcelcopter [8] (Figura 2.1b) sviluppato da DHL nel 2013 rappresenta la prima apparizione in Europa di un aeromobile senza pilota che effettuò un viaggio al di fuori del campo visivo del pilota in una missione reale. Le innovazioni tecniche del parcelcopter DHL includevano durata e autonomia di volo estese: La rotta di volo per l'isola di Juist è di circa 12 chilometri. Il volo come detto avvenne in modo completamente automatizzato per la prima volta. Per motivi di sicurezza e nel rispetto dei requisiti stabiliti dagli organismi responsabili, il parcelcopter DHL venne costantemente monitorato durante il volo da una stazione di terra mobile a Norddeich in modo da consentire un intervento manuale in tempo reale in caso di malfunzionamento o emergenza. Oltre a questo, la stazione di terra mantenne un costante contatto con i controllori del traffico aereo. Questo tuttavia non l'unico progetto su cui DHL lavorò, in quanto negli anni successivi vennero ideati due nuovi modelli, su cui l'azienda è ancora attiva, rispettivamente: il parcelcopter 3.0 e il parcelcopter 4.0.



(a) AeroTerrascan's drone Ai450.



(b) DHL's drone Parcelcopter.

Figura 2.1: Esempio dei due principali modelli di drone: (a) fixed-wing drone, (b) quadcopter.

Tuttavia esiste anche un ulteriore macro categoria definita dagli UAV **ibridi**, un tipo specifico di configurazione aerea che assoggetta i vantaggi delle precedenti due tipologie, ala fissa e ala rotante, avendo così la capacità di VTOL, velocità di volo elevate e maggiore distanza di volo percorribile. Questi dispositivi possono essere classificati in *convertiplano* e *tail-sitter*. La categoria convertiplano (Figura 2.2a) è rappresentata da un velivolo che genera portanza e spinta per mezzo di uno o più propulsori ad elica, di solito posti all'estremità di un'ala fissa. Esso combina la capacità di portanza verticale degli octocopter con la velocità e l'autonomia di un convenzionale fixed-wing. La tipologia di velivolo tail-sitter (Figura 2.2b), è caratterizzata dal fatto che il loro decollo e atterraggio avviene verticalmente sulla coda, inclinandosi in avanti per ottenere un volo orizzontale [21].



(a) Convertiplano (AW609).



(b) Tail-Sitter Hong Hu.

Figura 2.2: Esempio dei due principali modelli di drone ibridi: (a) convertiplano, (b) tail-sitter

## 2.2 Caratteristiche dei droni

Nella sezione precedente si è cercato di dare un quadro del complesso e vasto scenario connesso ai differenti modelli di drone attualmente sviluppati, mostrandone alcune delle innumerevoli ed eterogenee categorizzazione descritte in letteratura.

Sebbene esista una vasta gamma di dispositivi, alcune peculiarità tecniche, necessarie per valutare la loro idoneità rispetto a certe missioni, risultano essere comuni a tutti i modelli. Infatti è possibile osservare una serie di parametri e restrizioni comuni a tutti i diversi modelli di drone (Figura 2.3):

- **Specifiche del movimento:** i droni sono in grado di muoversi nello spazio 3D. Gli autopiloti dei droni sono solitamente in grado di mantenere la stabilità di volo, l'altitudine richiesta e gestire atterraggio e decollo in modo totalmente autonomo. Tuttavia, alcuni aspetti del movimento potrebbero dover essere argomento di analisi in considerazione della pianificazione delle operazioni da svolgere. Uno di questi è sicuramente la limitazione connessa all'angolo di virata minimo durante il volo, che è particolarmente importante per i droni ad ala fissa. Ad esempio, questa restrizione dovrebbe essere presa in considerazione durante l'irrorazione di terreni agricoli al fine di evitare punti non coperti. Tuttavia seppure droni di tipo rotocraft, come quadcopter e octocopter, non abbiano limitazioni di questo tipo, ogni inversione richiede tempo ed energia aggiuntivi, in quanto questo tipo di movimento richiede di fermarsi prima di muoversi nella direzione opposta. Pertanto, si cerca di calcolare traiettorie lineari con velocità costante limitando il raggio di sterzata anche per droni a rotore.
- **Carico limitato:** il carico è una componente determinante per i droni, soprattutto per quelli specializzati nella consegna pacchi. Normalmente questi dispositivi sono in grado di trasportare pacchi con una massa non superiore ai 3 kg, e di solito hanno capacità per il trasporto ristretta ad un solo pacchetto alla volta. Le limitazioni relative al carico sono strettamente legate alle proprietà energetiche della batteria utilizzata e alla dimensione e alla configurazione del drone. Ad esempio, per garantire un volo stabile, l'elica di un drone rotocraft dovrebbe generare una spinta sufficiente a vincere la forza di gravità. Pertanto, un drone più pesante ha un dispendio di energia maggiore di un drone più leggero pur percorrendo lo stesso tragitto.
- **Capacità di volo:** La maggior parte dei droni, ad eccezione di quelli alimentati direttamente da cavo, hanno un'energia limitata. Il consumo energetico di un drone dipende da una moltitudine di fattori, come il tipo (ala fissa o rotocraft), altitudine (in quanto, le eliche di un rotocraft devono ruotare più velocemente ad altitudini più elevate a causa della minore densità d'aria), condizioni di volo (come volo stazionario o in avanti), velocità di crociera, carico ed agenti atmosferici, che nello scenario studiato sono interpretati dalle condizioni ventose del luogo in cui ci si sta spostando. Questo parametro risulta essere uno

dei più significativi, infatti in relazione alle distanze che si riescono a percorrere, vengono definite le restrizioni dello scenario e di conseguenza le strategie risolutive per la missione da conseguire.

- **Connettività:** un ulteriore parametro da cui dipende la massima distanza percorribile è l'insieme delle tecniche e delle tecnologie applicate per la supervisione degli spostamenti del drone. Al fine di tutelarsi rispetto ad incidenti e situazioni problematiche durante il volo, i droni devono mantenere un costante collegamento con la stazione di controllo al fine di ricevere istruzioni e trasferire le informazioni raccolte. Poiché le comunicazioni avvengono normalmente attraverso una connessione Wi-Fi in *line-of-sight* (LOS), il segnale si potrebbe indebolire all'ombra degli edifici nelle aree urbane, a causa delle pareti se ci si sta spostando in uno spazio chiuso o rispetto ad ostacoli naturali. Si aggiunge a questo anche il fatto che le linee di trasmissione e le torri di telecomunicazione potrebbero causare interferenze di segnale. Pertanto, alcune aree la cui morfologia è particolarmente svantaggiosa potrebbero penalizzare o addirittura rendere impossibile il raggiungimento di una certa destinazione. Tuttavia, nel breve periodo è attesa una nuova regolamentazione per spostamenti di tipo *beyond-line-of-sight* (BLOS). Il volo BLOS può utilizzare le infrastrutture dedicate alla gestione del traffico aereo e le comunicazioni satellitari per mantenere il drone in contatto con i supervisori del traffico aereo, aumentando in questo modo la massima distanza percorribile.

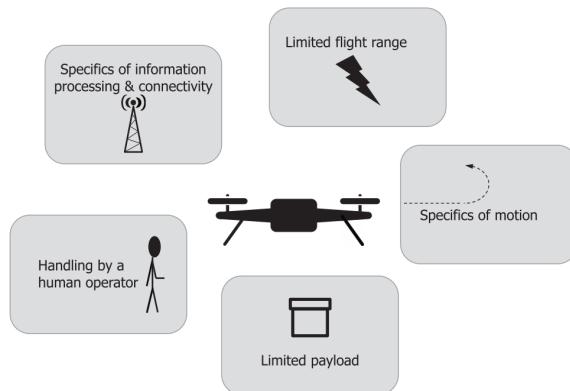


Figura 2.3: Schema riassuntivo dei diversi parametri di valutazione di un drone.

## 2.3 Applicazione dei droni in ambito civile

L’impiego dei droni in un contesto aziendale o più in generale civile potrebbe avere risvolti sia economici che ambientali molto interessanti. Infatti, come descritto nell’indagine [18], per molte realtà aziendali poter disporre di tecnologie pilotless rappresenterebbe un’apertura verso sostanziali risparmi sia in termini di forza lavoro che di risorse economiche. Oltre a questo, non si può esimere il fatto che i droni possono spostarsi e agire in ambienti particolarmente ostili o addirittura inaccessibili agli esseri umani, in quanto per loro natura non hanno la necessità di strade o infrastrutture per i loro spostamenti.

Oltre al mero risparmio economico, i consumi connessi al movimento di queste apparecchiature risulta essere maggiormente efficiente e di conseguenza eco-sostenibile rispetto agli odierni mezzi di trasporto pesanti alimentati a combustibile fossile. Un ulteriore vantaggio, intrinseco alla loro configurazione, è la rapidità con cui sono in grado di completare le missioni, peculiarità che li rende appetibili nel contesto delle consegne.

Proprio questo ambito, ma più in generale quello dei trasporti ha ricevuto recentemente un’attenzione considerevole da parte dei media, soprattutto grazie alle promettenti prospettive di consegna a basso costo. Le tipologie di consegna interessate potrebbero comprendere sia quelle a breve distanza all’interno di aree urbane e suburbane ma anche quelle in aree rurali localizzate in regioni remote. Oltre ai servizi di consegna, potrebbero trarre beneficio anche i servizi medico sanitari, che vedrebbero estendere la loro giurisdizione, mediante l’ausilio dei droni in grado di trasportare medicinali o addirittura sacche di sangue, anche in quelle aree in cui le infrastrutture stradali sono carenti o impraticabili durante determinati periodi dell’anno.

Tuttavia questo non è l’unico campo in cui l’impiego dei droni potrebbe appor-tare migliorie; infatti esistono ulteriori scenari altrettanto promettenti, come quello dell’agricoltura di precisione e quello del supporto operativo. Il settore agricolo è stato il primo campo in cui si è affermato l’utilizzo dei droni; la prima applicazione risale al 1980 quando vennero adoperati per trattare le risaie giapponesi con i pesticidi. Successivamente con lo sviluppo delle tecnologie connesse ai droni, si è riscontrato un crescente aumento delle applicazioni fra le quali il monitoraggio delle colture e il trattamento di precisione delle stesse. Il supporto operativo invece è un settore in via di sviluppo in cui l’utilizzo dei droni è congeniato al fine di supportare sia l’aspetto

strategico che la componente operativa di un qualsivoglia progetto. In questo contesto vengono sfruttati per controllare il procedere dei lavori di costruzione di una abitazione o per il supporto diretto nella fase di trasporto dei materiali all'interno di un cantiere.

Altre applicazioni [20], meno affermate rispetto alle precedenti, sono la gestione della sicurezza e della sorveglianza di un'area, mediante il lavoro coadiuvato fra droni e operatori, e l'utilizzo di droni in ambito cinematografico per il supporto allo sviluppo di effetti speciali, per riprendere scene e scattare fotografie.

## 2.4 Planning drone problems

Lo studio intrapreso all'interno di questo macro settore ha delineato un panorama particolarmente variegato dove, parallelamente alla nascita di nuovi modelli e nuovi domini di utilizzo, vi è stata una consistente crescita del numero di problemi correlati a questo campo. Questi appartengono alla famiglia dei *planning drone problems*, problemi di ottimizzazione combinatorica in cui uno o più droni devono raggiungere una o più postazioni dislocate in posizioni diverse di una mappa. Normalmente questi problemi vengono ricondotti al problema del *commesso viaggiatore* (TSP) [10], in cui si deve partendo da una certa sorgente attraversare tutti i punti di interesse presenti esattamente una volta per poi tornare alla sorgente. Tuttavia a seconda dell'obiettivo che si deve conseguire, si delineano differenti categorie di problema. Facendo riferimento a quanto riportato all'interno dell'articolo [20] è possibile suddividere i planning drone problems in relazione alle diverse tipologie di operazioni che si devono svolgere (Figura 2.4).

- Operazioni di copertura, dove uno o più droni devono coprire una determinata area utilizzando un sensore con gittata limitata.
- Operazioni di ricerca, dove l'obiettivo principe è individuare oggetti o edifici collocati in una certa area geografica.
- Operazioni di instradamento, dove uno o più droni devono attraversare un insieme finito di posizioni all'interno di una mappa.

- Operazioni di raccolta e trasferimento dati all'interno di una rete di sensori wireless, dove i droni devono ottenere una serie di informazioni dislocate all'interno di un insieme discreto di posizioni, considerando una serie di vincoli di comunicazione e di memoria.
- Operazioni di assegnazione di canali di comunicazione e potenza di calcolo a droni, in modo tale da garantire collegamenti fra i nodi della rete con un certo livello di qualità.

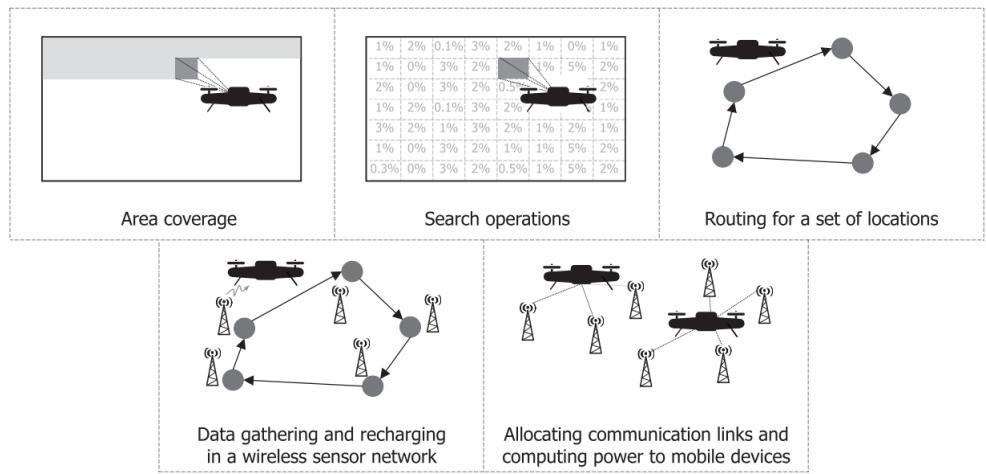


Figura 2.4: Classificazione delle diverse operazioni realizzabili dai droni.

#### 2.4.1 Problemi di copertura

Il *Coverage Path Planning* [4] (CPP) può essere classificato come una specializzazione del *motion planning problem* nella robotica, dove si deve costruire un cammino per un robot al fine di esplorare e coprire ogni area di un dato scenario. Questo può essere definito formalmente come: data un'area di interesse composta dallo spazio libero trafficabile dal drone e dai suoi confini, progettare un percorso che copre l'intero ambiente considerando le restrizioni di movimento e le caratteristiche del sensore installato sul velivolo, evitando però i possibili ostacoli. In un contesto aereo, gli ostacoli all'interno dello spazio trafficabile, che l'UAV non deve prendere in considerazione durante la fase di pianificazione, ad esempio, aree adiacenti ad aeroporti o edifici non significativi. L'ambiente target è di solito diviso in regioni non sovrapposte chiamate

celle, ognuna della quali definisce un singolo nodo all'interno del percorso pianificato. Queste vengono individuate utilizzando tecniche di decomposizione. Le dimensioni e la risoluzione delle celle possono variare a seconda della tecnica di decomposizione e in funzione di questo si dovrebbe applicare un'opportuna strategia al fine di garantire la loro copertura completa. Ovviamente, coprire una cella più grande potrebbe richiedere diversi movimenti, mentre per quelle più piccole potrebbe essere sufficiente anche un solo movimento. La dimensione di queste celle in genere è proporzionale alla portata del sensore o alla dimensione del drone. In questo contesto si distinguono principalmente i seguenti tipi di problemi.

- *Coverage path planning problem*, dove data una flotta di droni equipaggiati con sensori di portata limitata, prevede di visitare tutti le postazioni dislocate nell'area di interesse minimizzando il costo di attraversamento.
- *Partially coverage path planning problem*, dove data una flotta di droni con determinate restrizioni, l'obiettivo è quello di raccogliere il massimo quantitativo di informazioni dalla visita parziale dell'intera area.
- *Stationary coverage path planning problem*, dove data una flotta di droni, l'obiettivo è quello disporsi su postazioni fisse al fine di garantire la massima copertura possibile.

#### 2.4.2 Problemi di ricerca

I problemi di ricerca, sia quelli che utilizzano un singolo UAV sia quelli che impiegano una flotta di UAV, hanno l'obiettivo di cercare oggetti statici o dinamici all'interno di una certa area geografica. Questa categoria di problemi è stata particolarmente studiata nel corso del tempo e le prime ricerche si focalizzarono principalmente su metodi di ricerca locale. Quest'ultimi stabiliscono l'azione successiva valutando il più grande payoff o percorrendo cammini che evitano di lasciare grandi spazi non analizzati. Tali approcci sono semplici da implementare ed efficienti se considerati su piccola scala, ma spesso portano a scarse prestazioni di ricerca se valutati su larga scala. Studi più recenti hanno considerato la ricerca con l'utilizzo di droni come un problema di ottimizzazione globale, con l'obiettivo di minimizzare il tempo di rilevamento previsto o di massimizzare la probabilità di rilevamento accumulata

nell’intera operazione. Tuttavia, anche nel caso di un obiettivo stazionario con un singolo UAV, tale problema di ricerca è noto per essere NP-Hard [26], e quindi i metodi di ottimizzazione tradizionali richiedono tempi proibitivi per le istanze di grandi dimensioni. Inoltre, la maggior parte dei metodi richiede che sia nota a priori la distribuzione di probabilità associata alle posizioni che gli oggetti da individuare hanno all’interno dell’area di ricerca. Ciò nonostante, non è un compito banale stimare la distribuzione delle probabilità, e quindi può essere molto difficile sia adattare un metodo efficiente in un particolare scenario che in altri scenari simili o diversi.

Possiamo dunque sintetizzare un problema di ricerca come, data una mappa dell’area di ricerca  $P$  solitamente suddivisa in una collezione di celle (ad esempio, con una griglia) ciascuna etichettata con la probabilità di trovare un oggetto in quella cella, massimizzare la probabilità cumulativa di trovare l’oggetto entro un determinato lasso di tempo o, alternativamente, minimizzare il consumo di tempo per raggiungere la probabilità cumulativa desiderata di trovare l’oggetto.

### 2.4.3 Problemi di instradamento

In molte applicazioni sia di sorveglianza che di consegna, i droni devono eseguire un tour che attraversi una serie di posizioni, partendo e ritornando in un generico punto di partenza.

In questo contesto, i movimenti dei droni devono rispettare una mappa di routing che può essere modellata come un grafo in cui ogni arco (strada) ha un peso (tempo o consumo energetico) e ogni vertice (possibile obiettivo) ha una ricompensa. In questo modo è possibile rappresentare la mappa di routing di un certo edificio, città, regione o paese, in cui i droni effettuano i loro spostamenti. Uno schema in grado di modellare una generica applicazione potrebbe essere descritto come trovare un cammino da un punto di partenza a un punto di arrivo, in cui il punto di partenza, si suppone normalmente essere un magazzino da dove il drone inizia il volo, mentre la destinazione si considera essere la posizione del cliente o un determinato luogo che deve essere costantemente sorvegliato. Tuttavia, per i limiti precedentemente citati di batteria, il drone che attraversa i vari nodi della mappa ha la necessità di ricaricarsi. Quindi, si considera una missione end-to-end in cui il drone raggiunge la destinazione e torna al punto di partenza o di raccolta per potersi ricaricare. I routing problem richiedono quindi di trovare un cammino dal punto di partenza alla destinazione che

minimizzi alcuni parametri, come il tempo, la lunghezza o il consumo energetico del volo o in alternativa massimizzi la reward ottenuta attraversando i nodi del grafo.

Questa descrizione ci suggerisce la possibilità di modellare e risolvere i route planning problem come problemi di cammini minimi, o anche come versioni generalizzate di routing problem di base, come il TSP [10], il TSP multiplo [15], o il Vehicle Routing Problem [16].

#### 2.4.4 Problemi di raccolta e trasferimento dati

Prima di addentrarsi nella descrizione dei problemi di raccolta e trasferimento dati, è necessario precisare che il campo in cui questi vengono studiati è quello delle Wireless Sensor Network (WSN). Una rete di sensori wireless descrive una collezione di sensori wireless distribuiti nello spazio che raccolgono informazioni sull'ambiente circostante e le trasmettono alla stazione base. I droni all'interno di questo scenario fungono da strato aggiuntivo tra una rete di sensori stazionari e la stazione base: i sensori dislocati nell'ambiente raccolgono informazioni, e i droni raccolgono i dati dai sensori, trasmettendoli o riportandoli alla stazione base.

In una serie di applicazioni, la raccolta di dati da sensori stazionari assume una significativa importanza, ma senza l'impiego dei droni risulterebbe un'operazione lenta, proibitiva o addirittura impossibile. Pertanto, nuovi problemi che coinvolgono il routing dei droni nascono in concomitanza alla modellazione di WSN. Ad esempio, sott'acqua i gasdotti, alcuni dei quali di lunghezza superiore a 1000 km, sono monitorati da sensori stazionari posti periodicamente sulla loro superficie e spesso di difficile accesso. Un altro esempio è il monitoraggio dell'habitat degli animali selvatici, per il costante controllo delle specie in via di estinzione. Tuttavia una delle applicazioni più promettenti è l'individuazione dei sopravvissuti ad una catastrofe ambientale mediante l'utilizzo di una rete di sensori in grado di localizzare anche i segnali più deboli emessi dai telefoni cellulari. Ovviamente, in quest'ultimo caso, la raccolta di dati e la trasmissione di informazioni alla stazione di base dovrebbero avvenire con il minor ritardo possibile.

Questa preliminare introduzione, così come descritta, ricorda per certi versi l'insieme delle problematiche connesse al route planning. Tuttavia in contrasto con quanto osservato nella sezione precedente, le operazioni di raccolta dati dei droni nelle WSN deve rispettare vincoli di comunicazione, memoria e ricezione dei dati. Infatti

in questo contesto si hanno problematiche connesse alla limitata portata dei segnali utilizzati per le comunicazioni e un'affidabilità della trasmissione dei dati strettamente dipendente dalla distanza di comunicazione. Inoltre, a causa del limitato raggio di comunicazione, i nodi possono dover eseguire una trasmissione multihop (Figura 2.5). Nel campo delle telecomunicazioni una trasmissione diretta di informazioni da un nodo (sensore o drone) della WSN ad un altro nodo è chiamata hop.

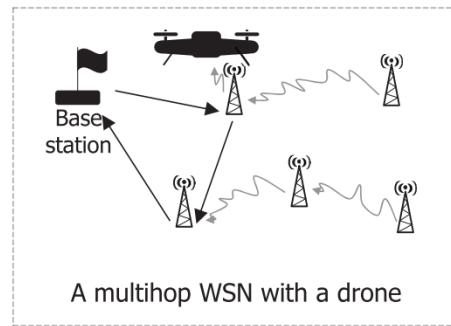


Figura 2.5: Schema di una rete WSN, in cui le frecce dritte rappresentano gli spostamenti del drone e le frecce ondulate raffigurano le comunicazione.

A seconda delle architetture di rete WSN implementate e delle restrizioni imposte sia dal modello di drone che dal tipo di sensore utilizzato, si delineano differenti scenari da studiare. Nel complesso però, poiché la raccolta e la trasmissione dei dati possiede delle restrizioni intrinseche alla natura dei dispositivi, come ad esempio la batteria limitata dei droni e dei sensori, si ha che alcune funzioni obiettivo sono più consuete rispetto ad altre. Quella maggiormente impiegata nella risoluzione dei problemi mira a massimizzare il *lifetime* della WSN, ossia l'intervallo temporale che intercorre dal primo guasto del sensore dovuto dall'esaurimento della sua energia. Si frappone la minimizzazione del massimo consumo energetico fra sensori e droni per trasmettere le informazioni raccolte. Tuttavia in letteratura si fa riferimento a tante altre funzioni obiettivo come ad esempio, la minimizzazione del *makespan*, della distanza totale percorsa dal drone e ancora del consumo energetico associato alle operazioni svolte dal drone.

## 2.4.5 Problemi connessi alle comunicazioni

Questa sottosezione si occupa di descrivere una famiglia di problemi la cui risoluzione individua una dislocazione dei droni, che permetta di estendere la copertura del segnale per la rete dei dispositivi mobili di una certa area geografica.

Per poter sfruttare la connettività offerta mediante l'uso di un drone, un dispositivo mobile dovrebbe essere situato all'interno del suo raggio di influenza. Pertanto, i problemi di pianificazione che ne derivano comportano decisioni sulla copertura dell'area. Tuttavia, a differenza dei problemi di pianificazione esaminati nella Sezione 2.4.1, queste decisioni riguardano la teoria della probabilità e le equazioni non lineari che descrivono i vincoli di comunicazione. Ad esempio, allocare canali di comunicazione con riferimento anche alle loro frequenze, al fine di estendere la copertura di rete per dispositivi mobili, selezionando la posizione dei droni in modo da ridurre al minimo le interferenze e garantire un rapporto segnale-rumore accettabile.

Esistono numerosi modi in cui sfruttare i droni. In primo luogo, si potrebbe pensare di assegnare i droni di cui si dispone a postazioni fisse, assumendo le sembianze di un ripetitore per i dispositivi mobili (Figura 2.6a). Sotto queste assunzioni si potrebbe ipotizzare inoltre che l'insieme delle posizioni da dover ricoprire venga specificato in anticipo. In secondo luogo, invece di rimanere in una posizione fissa, un drone potrebbe spostarsi, per esempio, lungo una traiettoria ciclica (Figura 2.6b). In terzo luogo, le trasmissioni drone-to-drone potrebbero simulare il comportamento di un ponte radio, ed essere sfruttate in modo che un dispositivo mobile collegato ad un drone possa stabilire collegamenti ad un dispositivo mobile collegato ad un altro drone (Figura 2.6c).

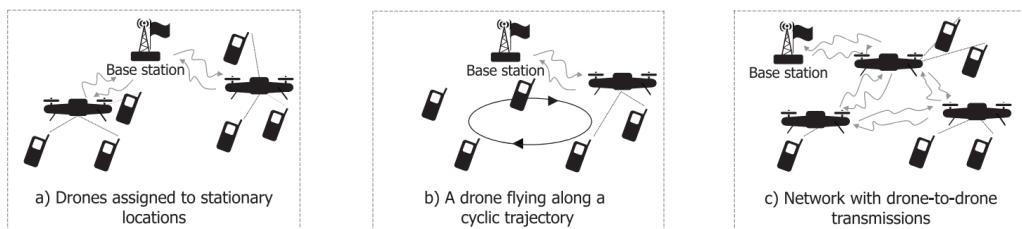


Figura 2.6: Illustrazione delle strategie di allocazione delle comunicazione per dispositivi mobili con l'ausilio dei droni.

Gli obiettivi più comuni in questo dominio di problemi riguardano la minimizza-

zione dei costi e la massimizzazione della qualità del servizio al cliente. Nel complesso il fornitore di servizi di comunicazione deve assegnare posizioni ai droni in modo che la connettività mobile desiderata sia conseguita con minimo costo o con il minimo numero di droni. Oltre a questo, se si considera che la trasmissione dei dati e le operazioni di collegamento richiedono un dispendio di energia per i droni, una funzione obiettivo comune potrebbe essere quella di massimizzare l'efficienza energetica o ridurre al minimo il dispendio di energia per raggiungere un livello accettabile di qualità della comunicazione. In alternativa, il numero di collegamenti di comunicazione stabiliti e la qualità del segnale, come ad esempio ritardi di trasmissione o probabilità di copertura, potrebbero essere elemento di ottimizzazione in funzione dei limiti di risorse e di energia dei droni.

# Capitolo 3

## Ammisibilità delle Missioni

In questo capitolo si introdurrà al *Mission Feasibility Problem* (MFP). Si tratta sostanzialmente di studiare la fattibilità di una missione (una consegna) tenendo in particolare conto che l'energia spesa dal drone dipende anche dalle condizioni dei venti locali. Si partirà con la descrizione del modello fisico che astrae la situazione ventosa dello scenario preso in esame. Si riporta uno studio che ha espresso il consumo energetico del drone come funzione della sua velocità, della velocità del vento, della direzione del vento, e del carico. Infine, si considera come instradare un drone da un punto all'altro. Nel corso dello studio svolto, sono stati proposti differenti tipologie di mappe di routing del drone. Verranno descritti gli aspetti teorici alla base dei differenti modelli di grafo costruiti e di seguito ne verranno illustrati i loro dettagli.

### 3.1 Mission feasibility problem

Si è osservato nel precedente capitolo che esistono numerose applicazioni che prevedono l'utilizzo dei droni. Fra i tanti scenari proposti si vuole porre l'attenzione verso l'utilizzo dei droni per effettuare consegne. Negli ultimi anni questo tipo di impiego ha avuto un crescente interesse e ha portato aziende del calibro di Amazon, Google, UPS e DHL ad interessarsi alla questione, tanto da integrare all'interno delle loro organizzazioni una sezione di sviluppo dedicata. L'utilizzo di tali dispositivi porta con sé numerosi lati positivi, come ad esempio sostanziali risparmi in termini economici e significative riduzioni delle emissioni prodotte.

Nello scenario preso in considerazione, lo schema che identifica le consegna può essere descritto come trovare un percorso da un punto di partenza a un punto di consegna. Per il punto di partenza, si supponga di avere un magazzino da dove il drone inizia il volo, e per il punto di consegna si fa corrispondere la posizione del cliente che può essere localizzato in qualunque punto della zona di competenza del magazzino. I movimenti dei droni devono rispettare una mappa di routing che può essere modellata attraverso un grafo in cui ogni arco (percorso) ha un peso (tempo o consumo energetico) e ogni vertice (posizione del cliente) ha una ricompensa. In questo modo è possibile rappresentare una mappa di routing di una qualsiasi area geografica.

In genere, per i limiti di carico, il drone può effettuare una sola consegna alla volta. Pertanto, la situazione presa in esame non può essere modellata come una generalizzazione del problema del commesso viaggiatore. A tal proposito, definiamo una consegna *end-to-end* in cui il drone raggiunge la posizione del cliente e torna al magazzino. Il problema di risolvere una consegna end-to-end in modo efficiente richiede di trovare un cammino dal magazzino al luogo di consegna che minimizzi alcuni parametri, come il tempo, la lunghezza o il consumo energetico del volo. Questa descrizione come osservato anche nella Sezione 2.4.3 identifica una particolare famiglia di problemi combinatorici, che in una loro versione base possono essere risolti calcolando lo *shortest-path* in un grafo pesato a partire da una certa sorgente. Ovviamente, la definizione di brevità dipenderà dal parametro che si intende minimizzare.

Un algoritmo classico per risolvere il problema end-to-end su di un grafo pesato è l'algoritmo di Dijkstra [6] applicato due volte: prima per andare dal magazzino al cliente finale, e poi dal cliente finale al magazzino. L'algoritmo di Dijkstra però presuppone che i pesi e la topologia del grafo siano dati e statici. Tuttavia la situazione in cui si sta lavorando prevede che i pesi e la topologia del grafo possano cambiare nel corso del tempo. Per esempio, i pesi degli archi, che modellano il tempo di volo e il consumo di energia, possono essere periodicamente perturbati da forze esterne. In particolare si assume senza perdita di generalità che l'insieme dei vertici e degli archi che descrivono l'area di consegna siano fissi, ma per quanto riguarda i pesi associati agli archi assumeranno dei valori diversi nel corso del tempo a seconda delle condizioni di vento. A tal proposito, per modellare i pesi del grafo in questione, si è deciso di costruire una nuova istanza dello stesso ad istanti regolari di tempo che

tenga in considerazione le forze applicate dal vento globale e relativo della zona. In quest'ottica avremo una serie di grafi o *snapshot* statici, con gli stessi archi e vertici, ma con pesi diversi a seconda della condizione ventosa corrente. Quanto detto è conosciuto in letteratura come *time-dependent graphs* [27] e così come citato in [23] si descriverà la sequenza di grafi osservati con il nome di *time-dependent cost delivery network graph*.

Riferendoci alle nozioni descritte in questa sezione, è dunque possibile dare una definizione del problema di ammissibilità della missione così come segue.

**Definizione(Mission-feasibility Problem).** Dato in input un time-dependent delivery network graph  $\mathcal{G} = \{\mathcal{G}_0, \dots, \mathcal{G}_t\}$ , un budget  $\mathcal{B}$  che rappresenta la carica massima della batteria del drone, un payload  $m_p$ , una velocità del drone  $v_d$ , e un vertice  $v$  come destinazione, il *Mission-Feasibility Problem* è un problema decisionale che richiede di stabilire se una certa missione sia fattibile o meno.

Quindi, il MFP ha come obiettivo quello di determinare se vi è un ciclo  $\mathcal{C}$  all'interno di  $\mathcal{G}$  che partendo dalla sorgente  $v_0$ , il magazzino aziendale, raggiunga la destinazione  $v$ , ovvero il cliente, e infine ritorni al nodo di partenza di costo inferiore o uguale al budget spendibile  $\mathcal{B}$  tenendo in considerazione le variazioni di consumi che si incontrano nel corso del tempo. Da un punto di vista pratico, risolvere il problema appena introdotto significa trovare due cammini rispettivamente, dal magazzino al cliente,  $\pi(v_0, v)$ , e dal cliente al magazzino,  $\pi(v, v_0)$  tali per cui  $d(\mathcal{C}) = d(\pi(v_0, v)) + d(\pi(v, v_0)) \leq \mathcal{B}$ . È importante però sottolineare che i cammini rispettivamente di partenza e di ritorno avranno differenti masse per il carico in quanto, raggiunto il vertice  $v$ , il cliente da servire riceve il pacco e quindi la successione di archi percorsi per il viaggio di ritorno verrà effettuata senza carico (i.e.,  $m_p = 0$ ).

## 3.2 Modellazione dei venti

Per poter studiare il MFP in uno scenario influenzato dalle forze dei venti, e per proporre una strategia risolutiva in riferimento allo stesso è necessario assumere, come già fatto nella sezione precedente, che l'insieme dei vertici e degli archi sia statico, pur avendo una funzione peso che descrive il consumo energetico che varia nel tempo.

Questo problema è stato studiato nella letteratura attuale con un approccio orientato alla teoria dei giochi, che risulta troppo lontano da un contesto reale perché

i costi degli archi non sono generici. Infatti, tale problema veniva modellato come una generalizzazione dell’*adversarial bandit problem* [2, 25] risolto sfruttando un processo decisionale di Markov in cui un agente si muove in un grafo aciclico con transizioni casuali. Questo determina uno scenario dove le ricompense sembrano locali, il grafo è aciclico e i pesi sono descritti da processi stocastici.

Invece di utilizzare variabili casuali per descrivere i cambiamenti climatici e il costo degli archi del grafo di riferimento, si propone di raccogliere le condizioni del vento quasi in tempo reale sfruttando le stazioni eoliche attraversate. Orientandosi in questa direzione si è deciso di sfruttare uno dei modelli fisici recentemente proposti in letteratura [24] che, dalla conoscenza della forza relativa e direzione del vento, il tipo di drone, la velocità del drone, l’altitudine del drone, e il carico, permette di desumere il consumo energetico per unità di lunghezza. Riferendosi a tali modelli, si propone di studiare il MFP, o più in generale il problema dello Shortest-Path in uno scenario in cui i pesi degli archi riflettono le condizioni ambientali real-time. Si descriveranno quindi i due modelli alla base della misurazione dell’energia spesa nel corso delle consegne, rispettivamente quello utilizzato per stimare la direzione relativa del vento e quello sfruttato per calcolare il consumo energetico.

### 3.2.1 Modello per i venti relativi

Una delle principali problematiche da risolvere, è descrivere un modello idoneo per definire la direzione del vento relativa alla strada che il drone sta percorrendo. Così come descritto in [23], si definisce il *vento globale*  $\omega$  come la forza ventosa che agisce nell’area di consegna che si sta attraversando. Si definisce poi  $\omega_s$  e  $\omega_d$ , rispettivamente la velocità e la direzione del vento globale  $\omega$ . All’interno dello scenario preso in esame, si avrà che i venti riferiti ad una certa area saranno sempre gli stessi, mentre le condizioni relative al vento globale potranno cambiare durante il corso del tempo.

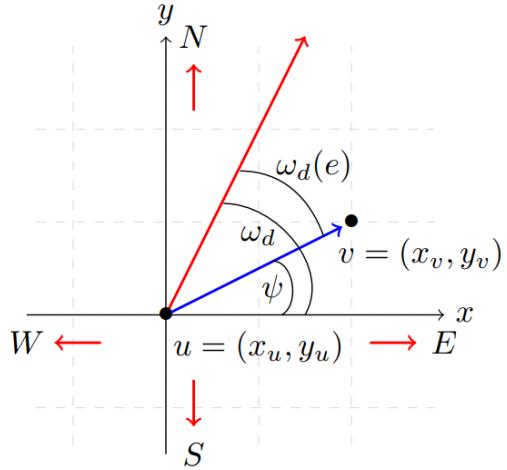


Figura 3.1: Direzione del vento relativo  $\omega_d(e)$ . Modello illustrativo in cui l'arco  $e$  è in blu e la direzione del vento globale è in rosso.

Pertanto, per definire la *direzione del vento relativo*  $\omega_d(e)$  percepita dal drone durante l'attraversamento dell'arco  $e$ , si costruirà un sistema di coordinate cartesiane con origine in  $u$  (vedi Figura 3.1) [23]. Quindi, sul segmento  $e = (u, v)$  attraversato dal drone, partendo dal punto  $u = (x_u, y_u)$  verso  $v = (x_v, y_v)$ , la direzione del vento relativo può essere calcolata come:

$$\omega_d(e) = \omega_d - \psi$$

dove  $\psi = \arctan\left(\frac{y_v - y_u}{x_v - x_u}\right)$  è la direzione dell'angolo associato all'arco  $e$ . Secondo quanto appena descritto, percorrere tratti con inclinazioni diverse, determina una direzione del vento relativo, percepito dal drone, differente. È conseguenza diretta, il fatto che la direzione dell'angolo relativa a due archi  $e = (u, v)$  ed  $e' = (v, u)$  differisca di  $180^\circ$ . Si assume inoltre che durante l'attraversata di un arco  $e$ , il vento globale rimanga stabile e quindi conseguentemente anche il vento relativo  $\omega_d(e)$  lo sarà.

Analogamente alle previsioni meteo che vengono riportate nei giornali, in questo lavoro verranno semplificate le possibili direzioni relative del vento, raggruppando i valori di  $\omega_d(e)$  in 8 settori (Figura 3.2a) che possono essere descritti dalla seguente ricorrenza.

$$[iL, (i+1)L] \quad \forall i = 0, \dots, 7, \quad \text{dove} \quad L = \frac{360^\circ}{8}$$

Gli otto settori vengono raggruppati due a due in quattro classi, rispettivamente  $C_0, C_1, C_2, C_3$  come segue:

$$C_i = C_i^U \cup C_i^L, \quad \text{dove} \quad C_i^U = [iL, (i+1)L], \quad C_i^L = [(7-i)L, (8-i)L] \quad \forall i = 0, \dots, 3$$

Per ogni classe, viene selezionato per definire l'angolo reale del vento relativo, l'angolo con il massimo coseno in valore assoluto. Tuttavia, siccome ridursi ai soli otto settori proposti in [23] potrebbe essere una soluzione eccessivamente povera, si è deciso di costruire un ulteriore modello che prevede però sedici settori (Figura 3.2b). La scelta di raddoppiare i settori, oltre che valida alternativa per il confronto dei risultati, delinea un approccio seppure discretizzato più aderente alla realtà.

La decisione di semplificare in settori le possibili direzioni del vento è stata varata al fine di discretizzare i possibili angoli relativi, e quindi per avere guadagni sia dalla diretta semplificazione dello scenario che dallo speed-up ottenuto costruendo un implementazione ad-hoc sotto queste condizioni.

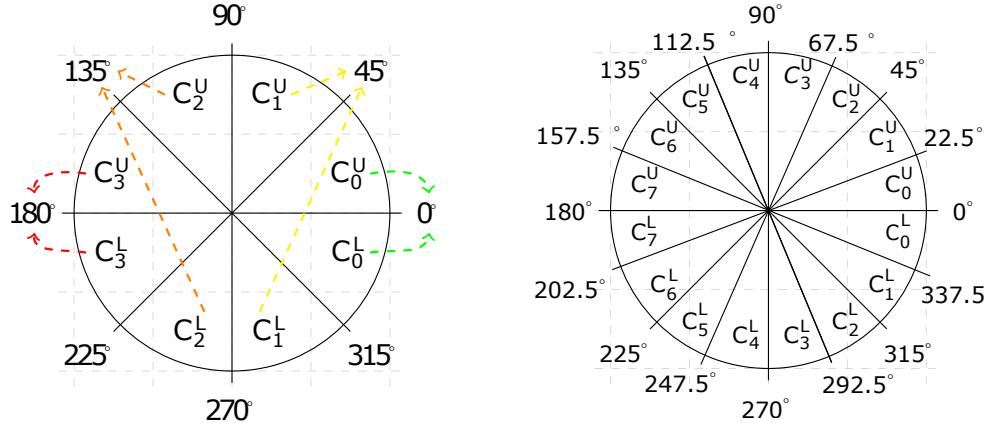


Figura 3.2: Schemi riassuntivi della semplificazione in classi della direzione del vento relativo  $\omega_d(e)$ .

### 3.3 Modello per i consumi energetici

Si è visto durante la trattazione che a seconda del modello di drone che si utilizza, si hanno consumi energetici differenti. Pertanto, al fine di definire la funzione peso del

grafo su cui si lavorerà si dovrà descrivere un modello fisico aderente alle specifiche tecniche e ambientali dello scenario. Nello specifico, rifacendosi alla letteratura [13] è stato derivato un modello che potesse descrivere il consumo energetico unitario di un octocopter con determinati parametri di peso, velocità e condizioni di vento locali [23].

Seppure nel corso del tempo alcuni parametri possano variare, la spinta totale  $T$  richiesta per far muovere il drone in avanti può essere così definita:

$$T = Wg + F_D, \quad (3.1)$$

dove  $W$  rappresenta il peso totale del drone che comprende anche la massa del carico che viene trasportato  $m_d$ ,  $g = 9,81m/s^2$  definisce la forza peso, e  $F_D$  è la forza di trascinamento totale. Il valore assunto da  $F_D$  dipende dalle proprietà del fluido e sulle dimensioni, forma e velocità del drone, e può essere stimato nel modo seguente:

$$F_D = \frac{1}{2}\rho v_a^2 C_D A, \quad (3.2)$$

dove  $\rho$  definisce la densità dell'aria,  $v_a$  è la velocità relativa del drone,  $A = \pi R^2$  è l'area occupata dal drone, e  $C_D$  è il coefficiente di resistenza. Il parametro  $v_a$ , ovvero la velocità relativa del drone, dipende a sua volta rispettivamente dalla velocità  $\omega_s$  e dalla direzione relativa  $\omega_d(e)$  del vento globale. Quindi la velocità  $v_a$  può così essere calcolata:

$$v_a = \sqrt[2]{v_N^2 + V_E^2}, \quad (3.3)$$

dove

$$\begin{aligned} v_N &= v_d - \omega_s \cos \omega_d(e) \\ v_E &= -\omega_s \sin \omega_d(e), \end{aligned} \quad (3.4)$$

dove  $v_d$  definisce la *velocità* del drone. Osservando le due componenti esplicitate (Eq. 3.4) è facile notare che quando la direzione del vento relativo  $\omega_d(e) = 0^\circ$ , si sta verificando la situazione ottimale in cui la direzione del vento globale e la direzione del drone in movimento hanno lo stesso orientamento. Questo significa che la componente  $v_E$ , ovvero l'attrito, è nulla e la componente  $v_N$  assume valore minimo, imponendo il valore minimo anche per  $v_a$ . Viceversa, quando  $\omega_d(e) = 180^\circ$ , si ha che la direzione del

vento e del drone sono opposte, con  $v_E$  nulla e  $v_N$  massima, e quindi  $v_a$  assume il valore massimo. Si noti che i due settori raggruppati in una singola classe nella Sezione 3.2.1 coprono gli stessi valori di velocità dell'aria. Infatti, poiché gli angoli dei due settori raggruppati condividono lo stesso coseno e hanno seno opposto, restituiscono lo stesso valore nell'Eq. 3.3.

Ricordando che l'angolo di inclinazione è  $\alpha = \arctan\left(\frac{F_D}{W_g}\right)$  [13], dopo aver calcolato il valore della spinta totale  $T$  dall'Eq. 3.1, è possibile calcolare la velocità indotta per librare  $v_h$  [13]:

$$v_h = \sqrt{\frac{T}{2\rho A}},$$

e da ultimo la velocità indotta  $v_i$  richiesta per soddisfare una determinata spinta  $T$  usando l'equazione implicita:

$$v_i = \frac{v_h^2}{\sqrt{(v_d \cos(\alpha))^2 + (v_d \sin(\alpha) + v_i)^2}}.$$

Infine è possibile stimare la potenza  $P$  per il volo stabile come segue [13]:

$$P = T(v_d \sin \alpha + v_i). \quad (3.5)$$

Infine, l'efficienza energetica,  $\mu(e)$ , richiesto per viaggiare lungo un arco di lunghezza unitarie  $e$  è calcolata come il rapporto tra il consumo di energia  $P$  e la velocità media a terra  $v_d$  del drone, vale a dire,

$$\mu(e) = \frac{P}{v_d}. \quad (3.6)$$

Pertanto, l'energia  $\eta(e)$  spesa per attraversare un arco  $e$  di lunghezza  $\lambda(e)$  può essere espressa come segue:

$$\eta(e) = \mu(e)\lambda(e). \quad (3.7)$$

Per completezza, la Figura 3.3 descrive l'energia spesa per attraversare un arco di lunghezza unitaria da un octocopter (un drone adatto per le consegne utilizzando i parametri dedotti da [24]) sotto diverse condizioni di vento reale registrate, carichi e velocità del drone. Si ha quindi che una maggiore massa del carico implica un

maggior consumo di energia, e spostarsi controvento piuttosto che a favore di vento rappresenta maggiori consumi di batteria.

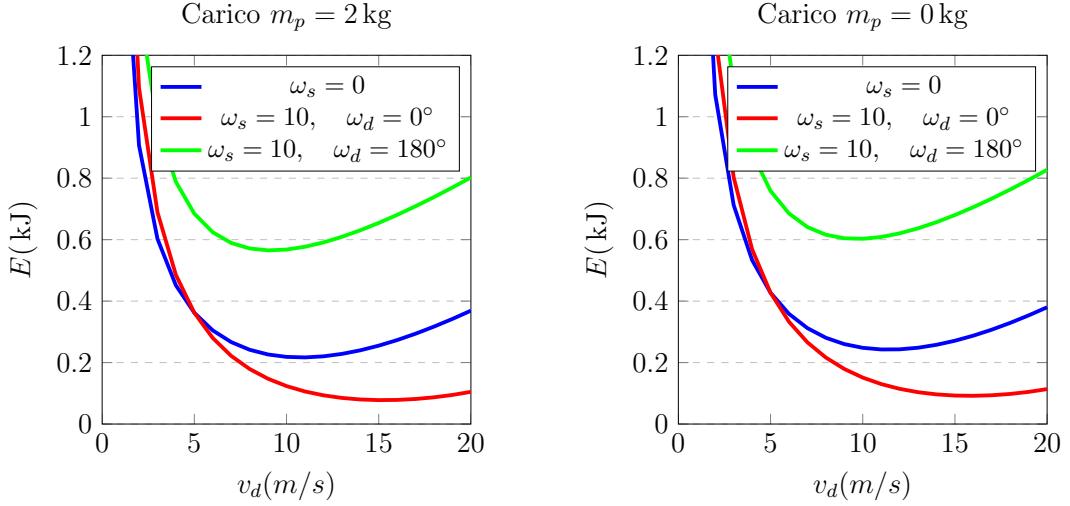


Figura 3.3: Grafico dei consumi energetici unitari di un drone con velocità media variabile  $v_d$ .

Se si analizzano i consumi energetici unitari facendo variare la direzione relativa del vento globale è possibile osservare che i massimi consumi si registrano quando il drone si sposta con un verso opposto a quello del vento globale e in corrispondenza della massima velocità del vento globale.

	$\omega_d = 0^\circ$	$\omega_d = 45^\circ$	$\omega_d = 135^\circ$	$\omega_d = 180^\circ$
$\omega_s = 0 \text{ m/s}$	0.22	0.22	0.22	0.22
$\omega_s = 5 \text{ m/s}$	0.15	0.17	0.32	0.36
$\omega_s = 10 \text{ m/s}$	0.12	0.18	0.50	0.55
$\omega_s = 15 \text{ m/s}$	0.15	0.23	0.74	0.84

Tabella 3.1: Tabella dei consumi unitari espressi in kJ, con  $m_p = 0 \text{ kg}$  e  $v_d = 0 \text{ m/s}$ .

La stessa cosa succede quando si ha un drone che trasporta con sé un carico di 2 kg, un carico realistico per un generico octocopter. Si ha dunque che in corrispondenza dell'aumento dell'angolo relativo si ha un incremento dei consumi unitari.

In tutte e tre le tabelle si ha che il consumo unitario minimo si riscontra nel momento in cui la velocità del drone corrisponde a quella del vento globale e l'angolo

	$\omega_d = 0^\circ$	$\omega_d = 45^\circ$	$\omega_d = 135^\circ$	$\omega_d = 180^\circ$
$\omega_s = 0 \text{ m/s}$	0.25	0.25	0.25	0.25
$\omega_s = 5 \text{ m/s}$	0.17	0.20	0.35	0.39
$\omega_s = 10 \text{ m/s}$	0.15	0.21	0.53	0.60
$\omega_s = 15 \text{ m/s}$	0.17	0.26	0.78	0.89

Tabella 3.2: Tabella dei consumi unitari espressi in kJ, con  $m_p = 2 \text{ kg}$  e  $v_d = 0 \text{ m/s}$ .

	$\omega_d = 0^\circ$	$\omega_d = 45^\circ$	$\omega_d = 135^\circ$	$\omega_d = 180^\circ$
$\omega_s = 0 \text{ m/s}$	0.33	0.33	0.33	0.33
$\omega_s = 5 \text{ m/s}$	0.25	0.28	0.44	0.47
$\omega_s = 10 \text{ m/s}$	0.23	0.29	0.62	0.70
$\omega_s = 15 \text{ m/s}$	0.25	0.34	0.88	1.00

Tabella 3.3: Tabella dei consumi unitari espressi in kJ, con  $m_p = 7 \text{ kg}$  e  $v_d = 0 \text{ m/s}$ .

relativo è di  $0^\circ$ . Questo succede perché avendo un vento a favore il drone non ha la necessità né di accelerare né di frenare e quindi è in grado di trarre il massimo risparmio energetico sfruttando le suddette condizioni di vento globale. Si può inoltre osservare che con l'aumento del peso del carico che il drone trasporta si ha un conseguente aumento dei consumi anche se tuttavia l'incremento sia fra le tre tabelle che all'interno delle singole non ha un passo lineare.

## 3.4 Modellazione dello scenario

Negli ultimi anni è stato registrato un considerevole aumento dei dispositivi installati all'interno delle città per il controllo delle situazioni meteorologiche e dell'inquinamento atmosferico delle stesse. Questi dispositivi molto spesso identificati come colonnine meteorologiche, possono avere differenti ruoli all'interno del contesto cittadino. Si hanno infatti tecnologie volte alla produzione di energia rinnovabile, come ad esempio pale eoliche e pannelli solari, ma anche apparecchiature multi-funzionali, come ad esempio gli Slaap [19], che stanno arricchendo e trasformando l'architettura sia tecnologica che urbanistica delle città. La crescita del numero delle smart-city porta con sé migliorie esplicite, connesse ai servizi direttamente offerti dalle tecnologie installate ma anche vantaggi impliciti, come la raccolta e il salvataggio di informazioni di diversa

natura, l'aumento delle connessioni wireless e ancora il monitoraggio dell'ambiente circostante.

Fra le molteplici potenzialità delle smart-city vi è la possibilità di funzioni di supporto per i droni, come ad esempio funzioni di ricarica o anche semplicemente di base di decollo e atterraggio [19].

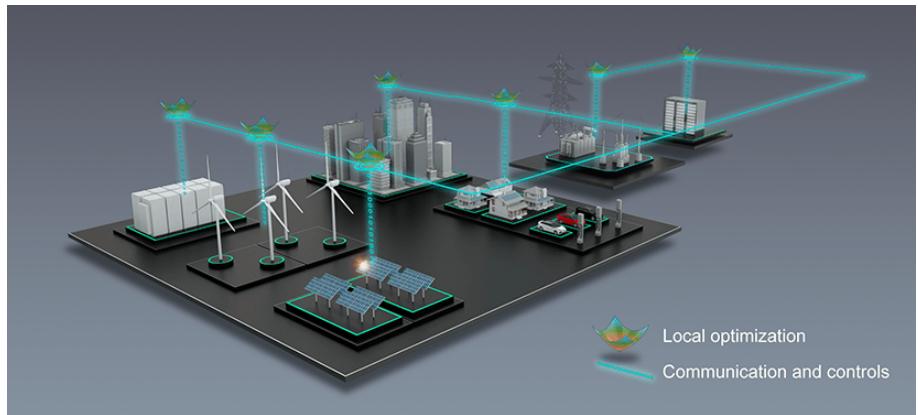


Figura 3.4: Rappresentazione di un possibile prospetto di smart-city.

Tuttavia, limitandosi per così dire alla prima funzione citata, ovvero quella di stazione meteorologica, si potrebbe definire in funzione di questo un modello di studio su cui analizzare sia le rotte che i droni potrebbero intraprendere che i consumi energetici ad esse associate. In questa sezione verranno dunque descritte in linea generale le diverse strategie con cui si è pensato di descrivere il modello di riferimento per il routing dei droni, tutto questo con il fine di definire l'instradamento del drone fra due punti, basandosi su concetto di area di "competenza" di una centraline. L'idea è di associare ciascun punto del piano alla centralina più vicina e di collegare poi le due centraline o 'in modo diretto' (vedi Sezione 3.4.2) o passando per vertici aggiunti (vedi Sezione 3.4.1). La trattazione di questi argomenti verrà preceduta da una illustrazione dei concetti teorici su cui si fondono i differenti approcci costruiti.

### 3.4.1 Tassellazione di Voronoi

Il diagramma di Voronoi [9] è una struttura dati ampiamente studiata nel campo della geometria computazionale. In origine, caratterizzava le regioni di prossimità per un

insieme di  $k$  siti nel piano dove la distanza dei punti è definita dalla loro distanza euclidea.

In matematica, un diagramma di Voronoi è una partizione di un piano in regioni vicine ad un certo insieme di elementi. Nel caso più semplice, questi elementi sono punti nel piano (chiamati semi, siti, o generatori). Per ogni seme c'è una regione corrispondente costituita da tutti i punti del piano più vicini a quel seme che a qualsiasi altro. Queste regioni sono chiamate celle di Voronoi.

Nel caso più semplice, mostrato nella Figura 3.5, viene fornito un insieme finito di punti  $p_1, \dots, p_n$  nel piano euclideo. In questo caso ogni sito  $p_k$  è semplicemente un punto, e la sua corrispondente cella di Voronoi  $R_k$  è costituita da ogni punto del piano euclideo la cui distanza da  $p_k$  è inferiore o uguale alla sua distanza da qualsiasi altro  $p_k$ . Ciascuna di queste celle è ottenuta dall'intersezione di semispazi, e quindi sottendono un poligono convesso. I segmenti che definiscono i bordi del diagramma di Voronoi sono tutti i punti nel piano che sono equidistanti ai due siti più vicini, mentre per quanto riguarda i vertici di Voronoi (nodi) sono tutti i punti equidistanti a tre (o più) siti.

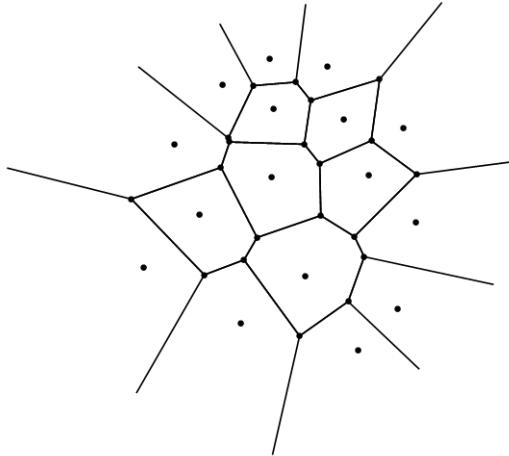


Figura 3.5: Tassellazione di Voronoi di un insieme di punti  $P$ .

**Definizione(Tassellazione di Voronoi).** Sia  $X$  uno spazio metrico con una funzione distanza  $d$ . Sia  $K$  un insieme di indici e sia  $(P_k)_{k \in K}$  una tupla (collezione ordinata) di sottoinsiemi non vuoti di siti nello spazio  $X$ . La cella di Voronoi, o

regione di Voronoi,  $R_k$ , associata al sito  $P_k$  è l'insieme di tutti i punti in  $X$  le cui distanze da  $P_k$  non sono più grandi della loro distanza rispetto ad altri siti  $P_j$ , dove  $j$  un generico indice diverso da  $k$ .

Nel panorama delle procedure risolutive vi sono numerose soluzioni efficienti per la costruzione dei diagrammi di Voronoi. Queste sulla base dell'approccio risolutivo applicato possono essere classificate come: procedure dirette, dall'insieme dei punti si calcola *direttamente* la tassellazione ad essi associata, e procedure indirette, in cui partendo da una struttura, come ad esempio la triangolazione di Delaunay, si deriva la corrispondente tassellazione di Voronoi.

Uno degli algoritmi diretti più famosi è l'algoritmo di Fortune [11] il quale restituisce un tassellazione valida con una complessità temporale di  $\Theta(n \log n)$  e con complessità spaziale di  $\mathcal{O}(n)$ . Questo algoritmo utilizza durante la sua esecuzione due iperpiani, uno chiamato *sweep line* che definisce una retta e un altro chiamato *beach line* che definisce una complessa figura piana paragonabile ad una parabola, i quali si muovono sul piano con l'avanzare dell'algoritmo.

Questo può essere descritto in modo informale come segue. Si noti che alcuni concetti, nella descrizione riportata, sono stati semplificati per permettere una comprensione a grandi linee dell'algoritmo. La sweep line per convenzione si suppone sia orientata verticalmente e che si muova da sinistra verso destra. Durante l'esecuzione dell'algoritmo viene garantita la seguente invariante di ciclo, ovvero i punti forniti in input di cui si deve costruire il corrispondente diagramma di Voronoi che si trovano a sinistra della sweep line sono già incorporati nel diagramma di Voronoi, mentre i punti a destra della stessa devono ancora essere processati. La beach line individua la parte del piano entro la quale può essere definita una cella di Voronoi, indipendentemente dai punti che si trovano alla destra della linea di sweep. Per ogni punto a sinistra della linea di sweep, è possibile definire una parabola di punti equidistanti rispetto al punto preso in considerazione e alla sweep line stessa; la beach line sancisce quindi il confine di queste parbole. Man mano che la linea di scansione progredisce, i vertici della beach line, su cui si incrociano due parbole, tracciano i bordi del diagramma di Voronoi. La beach line progredisce mantenendo la base di ogni parabola esattamente a metà strada tra i punti inizialmente individuati dalla linea di sweep, e la nuova posizione della stessa. In questo modo si ottiene un diagramma di Voronoi con un approccio incrementale che aggiunge una cella ogni qualvolta incontra uno dei punti

forniti in input.

Per quanto riguarda le procedure indirette, la più consueta è quella di usare l'algoritmo Bowyer-Watson [17] come sub-routine e poi dal risultato parziale derivarne il diagramma di Voronoi. Questa costruisce in modo efficiente la triangolazione di Delaunay associata ai punti dati in input e una volta fatto questo viene derivato il diagramma di Voronoi corrispondente calcolando il grafo duale della triangolazione in tempo lineare rispetto al numero di punti che viene dato in input.

I diagrammi di Voronoi hanno applicazioni pratiche e teoriche in molti campi, principalmente nella matematica e nell'informatica, ma anche nell'arte visiva. Le applicazioni più importanti nel campo informatico sono le seguenti:

- Nella teoria delle reti, i diagrammi di Voronoi possono essere utilizzati per derivare le capacità di una rete wireless [9].
- Nella computer graphics, i diagrammi di Voronoi sono usati per calcolare la 3D shattering o noti anche come modelli di geometria di frattura. È anche usato per generare in modo procedurale texture organiche [5].
- Nella navigazione autonoma di robot, i diagrammi di Voronoi sono usati per trovare percorsi liberi da ostacoli. Se si considerano come centri gli ostacoli, allora gli archi del grafo saranno le vie più lontane dagli ostacoli (e teoricamente eventuali collisioni) [9].
- Nel machine-learning, i diagrammi di Voronoi sono usati per fare le classificazioni 1-NN [9].

### 3.4.2 Triangolazione di Delaunay

La triangolazione di Delaunay prende il nome da Boris Delaunay per il suo lavoro [7] su questo argomento del 1934. In matematica e geometria computazionale, una triangolazione di Delaunay (nota anche come triangolazione di Delone) per un dato insieme  $P$  di punti discreti in un piano è una triangolazione  $DT(P)$  (vedi Figura 3.6) tale per cui nessun punto in  $P$  si trova all'interno del circoncerchio di qualsiasi triangolo in  $DT(P)$ .

**Definizione(Circoncerchio).** Un circumcerchio è la circonferenza circoscritta a un triangolo, ovvero l'unica circonferenza passante per tutti i suoi tre vertici, il cui centro è detto circocentro e il raggio circumraggio.

L'obiettivo principale della triangolazione di Delaunay è quello di massimizzare il minimo angolo di tutti gli angoli dei triangoli nella triangolazione o di minimizzare il massimo circumraggio per ogni elemento.

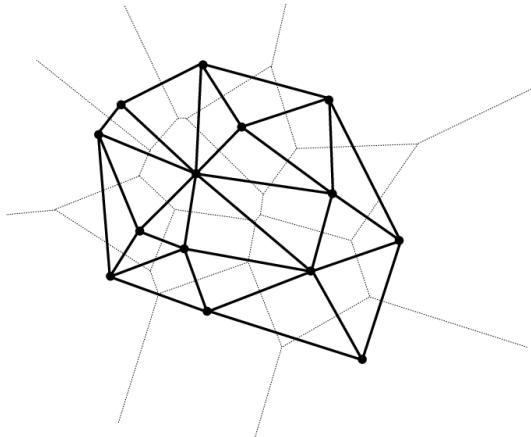


Figura 3.6: Triangolazione di Delaunay di un insieme di punti  $P$ .

La triangolazione di Delaunay di un insieme di punti discreti  $P$  in posizione casuale corrisponde al grafo duale del diagramma di Voronoi per  $P$ . I circoncentri dei triangoli di Delaunay sono i vertici del diagramma di Voronoi. Nel caso bidimensionale, i vertici di Voronoi sono collegati tramite archi, che possono essere derivati dalle relazioni di adiacenza dei triangoli di Delaunay: Se due triangoli condividono un arco nella triangolazione di Delaunay, i loro circoncentri devono essere collegati con un bordo nella tassellazione di Voronoi.

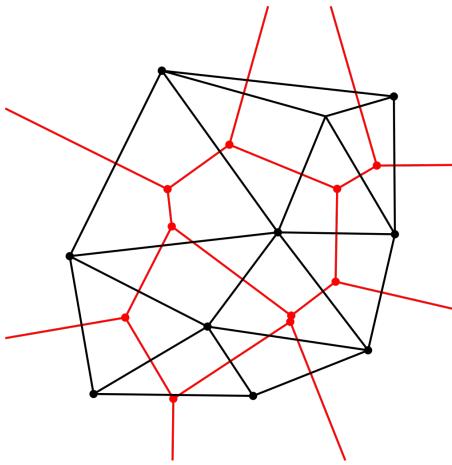


Figura 3.7: Collegando i centri dei circumcerchi si ottiene il diagramma di Voronoi (in rosso).

Grazie al risultato di Shamos e Hoey [22] è stato dimostrato che la triangolazione di Delaunay [17] di un insieme di  $n$  punti nel piano può essere costruita in tempo  $\mathcal{O}(n \log n)$ , che è asintoticamente ottimale. Gli stessi autori per poter dimostrare un tale limite asintotico, proposero un algoritmo per la costruzione di una triangolazione di Delaunay il quale veniva implementato con il paradigma *divide et impera*. Questo algoritmo che risulta essere il più efficiente fra quelli attualmente conosciuti, partiziona in insiemi di al più tre punti i punti che riceve in input. Dopodiché la costruzione dei triangoli avviene eseguendo il merge fra gli insiemi precedentemente calcolati in modo che la triangolazione in ogni istante sia legale; se il circumcerchio costruito su di un triangolo contiene un ulteriore punto al suo interno allora significa che quel triangolo è contiene un lato illegale.

Un altro esempio di algoritmo che risolve il problema della triangolazione di Delaunay è il precedente citato algoritmo di Bowyer-Watson [17]. Questo algoritmo, per la sua natura, è particolarmente efficiente quando la costruzione della triangolazione deve avvenire in modo incrementale. Il suo funzionamento si basa ancora una volta sul concetto di triangolazione legale, in quanto durante ogni ciclo di computazione viene aggiungendo progressivamente un nuovo punto causando una conseguente alterazione della triangolazione corrente che deve rimanere però una triangolazione legale. Quindi dopo ogni inserzione, tutti i triangoli i cui circumcerchi contengono il nuovo punto vengono cancellati, lasciando un foro poligonale a forma di stella che viene poi ri-triangolato utilizzando il nuovo punto. Utilizzando particolari tecniche

per la localizzazione dei triangoli da rimuovere, l'algoritmo appena descritto ha una complessità temporale di  $\mathcal{O}(n \log n)$  dove  $n$  è la cardinalità dell'insieme dei punti dati in input, anche se esistono casi degeneri speciali la complessità cresce fino a diventare  $\mathcal{O}(n^2)$ .

Le triangolazioni di un insieme di punti nel piano sono state ampiamente studiate e hanno applicazioni nel *closest point problem*, nel metodo degli elementi finiti, nell'analisi delle sollecitazioni dei continui bidimensionali e nell'interpolazione. Dal punto di vista pratico, questo metodo viene adoperato in differenti applicazioni:

- Le triangolazioni di Delaunay possono essere utilizzate per determinare la densità o l'intensità dei campionamenti di punti mediante la Delaunay tessellation field estimator (DTFE).
- Le triangolazioni di Delaunay vengono utilizzate per modellare il terreno o altri oggetti caratterizzati da una serie di punti di campionamento.
- La *constrained Delaunay triangulation* ha trovato applicazioni nella pianificazione del percorso nella guida automatizzata.

Nel seguito si descriverà come questi concetti appena visti verranno usati nell'instradamento dei droni, in particolare per la costruzione del grafo denominato come "Delaunay-oriented".

### 3.4.3 Grafo Voronoi-Oriented

Il grafo denominato come "Voronoi-oriented" è stato definito al fine di delineare vie percorribili per i droni, tali per cui i venti che le influenzano sia ben definiti. L'idea di utilizzare i diagrammi di Voronoi per modellare lo scenario che si sta analizzando è nata in relazione alla definizione stessa di tassellazione di Voronoi (def. 3.4.1). La tassellazione di Voronoi dato un insieme di punti disposti nel piano, descrive un insieme di celle costruite a partire da essi tali per cui le distanze dei punti racchiusi al loro interno sono minimizzate rispetto al centro della cella. Traslando la nozione puramente geometrica in un contesto reale, è possibile, costruendo una tassellazione di Voronoi a partire dai punti individuati dalle coordinate spaziali delle centraline, definire le aree di influenza di un certo vento. Nello specifico la regione di Voronoi

individuata da una particolare centralina, delinea un area geografica assoggettata dalle condizioni ventose rilevate dalla stessa.

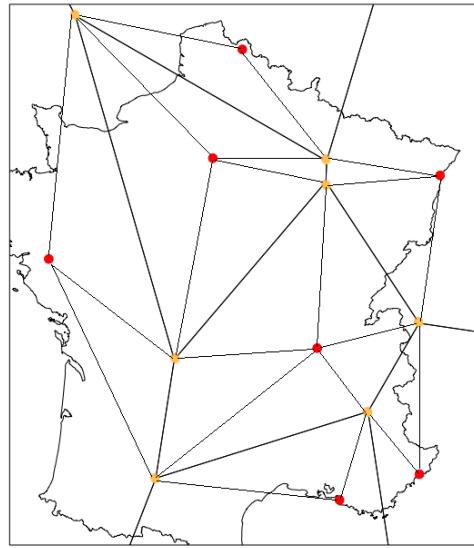


Figura 3.8: Illustrazione di un grafo Voronoi-Oriented.

Disponendo ora di un partizionamento delle aree geografiche assoggettate dalla stessa tipologie di vento, è possibile definire un insieme di percorsi utilizzabili dai droni per potersi spostare. Per gli studi che si dovranno fare si ha la necessità di descrivere un grafo che simuli l'area geografica su cui si dovranno effettuare le consegne. Pertanto, si costruirà un grafo il cui insieme degli archi  $E$  sarà composto da tutti i lati che delimitano le regioni di Voronoi e dall'insieme degli archi che congiungono il centro ai vertici di ogni regione di Voronoi. In quest'ultimo caso, gli archi che giacciono all'interno di una cella di Voronoi vengono perturbati da un solo e ben definito vento. Pertanto, sfruttando il modello fisico costruito sarà possibile stabilire il consumo energetico su questi archi in modo diretto (Figura 3.3), utilizzando la seguente formula,  $\eta(e) = \mu(e)\lambda(e)$ .

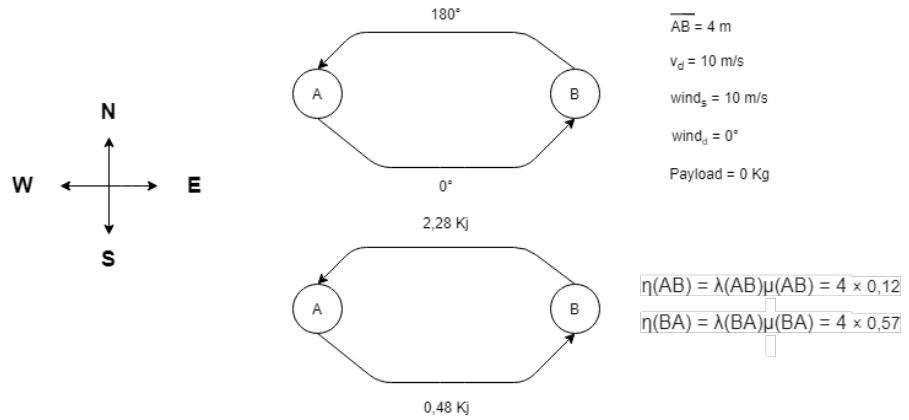


Figura 3.9: Esempio di calcolo dei consumi di un grafo generico.

Per quanto riguarda gli archi che rappresentano i lati delle regioni di Voronoi, questi si trovano equidistanti dal al più due centri; i lati di una regione di Voronoi possono infatti essere lati contesi da più di una cella e per questa ragione la condizione ventosa non è chiara e ben definita come nel precedente caso. La situazione ventosa di questi archi viene determinata con la seguente strategia, a seconda della direzione che il drone possiede quando attraversa un tale arco subirà l'effetto del vento la cui area di influenza(centralina) si trova alla sua destra. Sfruttando la cosiddetta *regola della mano destra* è possibile determinare il vento che influenza la tratta che si sta percorrendo. Così facendo si avrà che gli archi posizionati agli estremi delle celle di Voronoi avranno una certa condizione ventosa se attraversati da sinistra verso destra mentre se attraversati da destra verso sinistra subiranno l'influenza di un altro vento. L'insieme dei vertici  $V$  è caratterizzato invece da tutti i vertici delle regioni e dei centri di Voronoi.

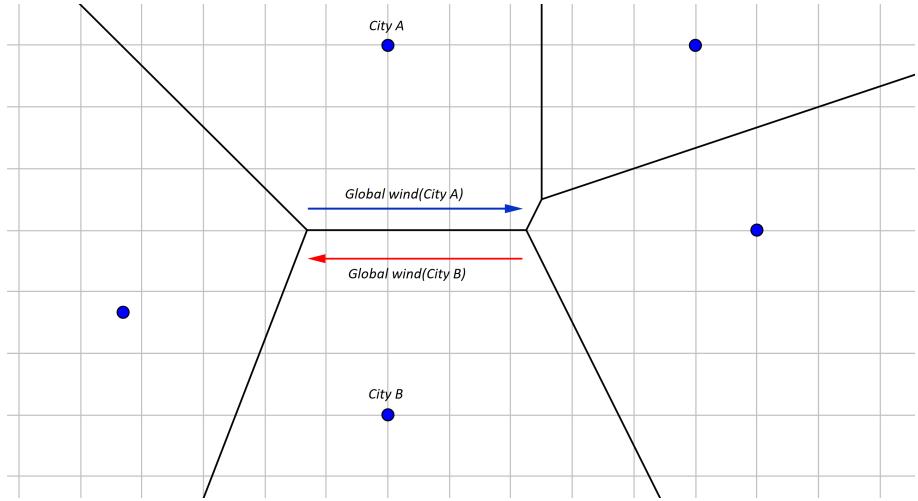


Figura 3.10: Esempio di associazione dei venti su un lato di una regione di Voronoi.

La soluzione così come proposta non è mai stata descritta in letteratura e per questa ragione non esiste un fattore approssimazione per quanto riguarda le distanze rispetto ad un grafo Euclideo. Tuttavia, considerando una stima del tutto empirica delle distanze, si può avanzare l'ipotesi che il fattore che approssima i cammini minimi di un grafo Voronoi-Oriented rispetto a quelli di un grafo Euclideo sia piuttosto cospicuo.

Sebbene questa osservazione sia poco incoraggiante, è possibile notare, così come descritto in [3], che in un contesto reale dove vi è la presenza di ostacoli la modellazione prodotta sfruttando la nozione di tassellazione di Voronoi risulta essere maggiormente performante rispetto alla semplice valutazione della distanza Euclidea. Infatti se si costruisce una mappa di Voronoi sulla base dei punti individuati dagli ostacoli nello spazio, si è in grado di muoversi senza la necessità di ulteriori computazioni sui lati delle regioni che descrivono la tassellazione, in quanto tali segmenti rappresentano i punti più distanti fra i più vicini dal centro, ovvero gli ostacoli. Se si lavorasse invece in un grafo Euclideo si avrebbe la necessità di analizzare istante per istante lo scenario che si sta attraversando e in corrispondenza di un ostacolo ricalcolare il percorso da seguire per raggiungere la destinazione.

Oltre a quanto detto vi è un ulteriore vantaggio non trascurabile dovuto dall'utilizzo di questo tipo di modello, infatti rappresenta una natura tecnica per definire la topografia ventosa di una determinata area geografica o più in generale di un qualsivoglia scenario reale.

### 3.4.4 Grafo Delaunay-Oriented

Si è descritta in precedenza la nozione di triangolazione di Delaunay e si è osservato che, dato un insieme di punti, questa rappresenta il grafo *duale* del corrisponde grafo di Voronoi. Se si osservano con la lente di ingrandimento i dettagli associati alla relazione fra grafo di Voronoi e grafo della triangolazione di Delaunay si ha che ogni lato di Voronoi condiviso fra due celle è attraversato da un arco di Delaunay. Pertanto, rifacendosi a quanto detto per il grafo Voronoi-Oriented si ha che ogni cella confina una certa situazione ventosa, dunque si ha che gli archi di Delaunay che attraversano esattamente due celle sono influenzate da venti ben definiti.

Per ogni lato di Voronoi si ha un corrispondente lato di Delaunay [7]. Tuttavia, seppure alcuni archi della triangolazione attraversano esattamente due celle di Voronoi non è condizione necessaria che tutti gli archi verifichino tale proprietà. Pertanto non tutti gli archi che appartengono alla triangolazione hanno condizioni di vento ben definite (vedi Figura 3.11).

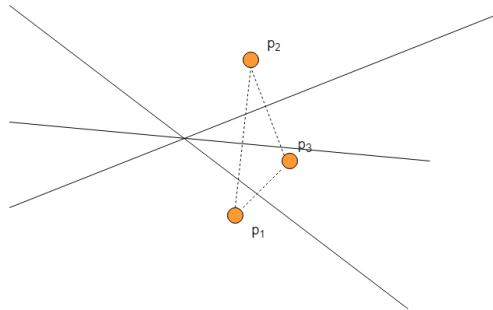


Figura 3.11: Esempio di un'istanza di grafo in cui i lati di Delaunay attraversano più di due regioni di Voronoi.

Questo accade ogni qualvolta le bisettrici di tre regioni di Voronoi si intersecano in uno stesso punto che individua il centro della circonferenza che inscrive il triangolo della triangolazione corrispondente.

La costruzione del grafo di instradamento del drone basato su Delaunay prevede perciò di selezionare sin da subito tutti gli archi appartenenti alla triangolazione di Delaunay costruita sulla base delle stazioni meteo di cui si dispone che attraversano solo ed esclusivamente due celle di Voronoi. Gli archi che non soddisfano tale proprietà vengono eliminati e sostituiti da due archi che si appoggiano su di un vertice di

Voronoi. Quest'ultimi congiungono due centraline ad un vertice di Voronoi, il quale minimizza la distanza fra i due centri posizionati come estremi opposti dei due archi.

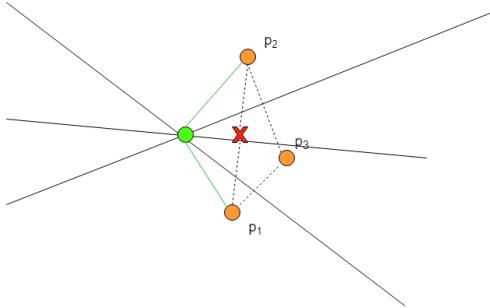


Figura 3.12: Esempio di un'istanza di soluzione in cui i lati di Delaunay attraversano più di due regioni di Voronoi.

Le condizioni di vento associate agli archi non appartenenti alla triangolazione sono ben definite e per queste valgono le stesse regole descritte per il grafo Voronoi-oriented. Per quanto riguarda gli archi di Delaunay selezionati come appena descritto, questi vengono assoggettati da due venti, ovvero quelli circoscritti dalle regioni di Voronoi che racchiudono i due centri. Inoltre, per costruzione tutti gli archi di Delaunay, ossia quelli che attraversano due celle di Voronoi, sono perpendicolari e dividono a metà il lato di Voronoi che attraversano [7]. Quindi anche per questi archi il calcolo dei consumi energetici può essere effettuato come nel caso precedente, tenendo in considerazione il vento di ciascun estremo per metà dell'arco, come mostrato in Figura 3.13. In Figura 3.13 si mostra un caso estremo, tuttavia, possibile. È chiaro che si tratta di una 'rozza' semplificazione: il cambiamento del vento non sarà mai istantaneo né nello spazio né nel tempo. Tuttavia, è una semplificazione compatibile con uno studio di ammissibilità.

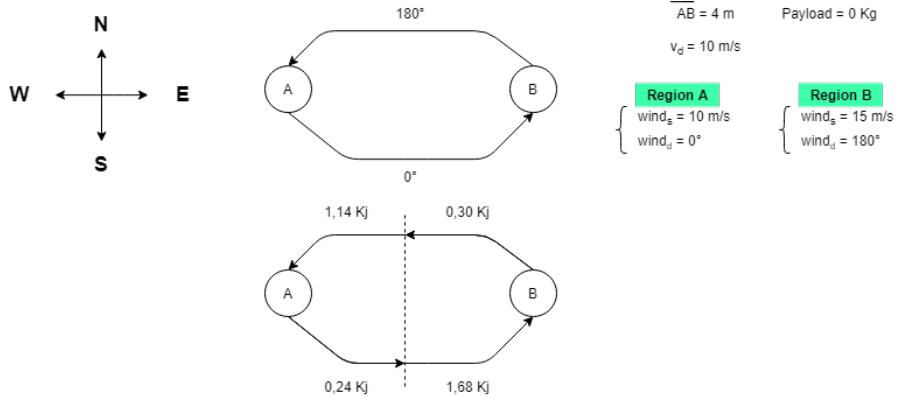


Figura 3.13: Esempio di associazione dei venti su un lato della triangolazione di Delaunay.

Il grafo così costruito è molto più sparso di quello 'Voronoi-oriented' discusso nella precedente sezione. Un grafo sparso ha un numero limitato di cammini da punto-a-punto, ossia l'instradamento del drone sarà quasi univoco. Tuttavia se si analizzano le lunghezza dei cammini minimi, queste risultano essere una approssimazione della distanza Euclidea. In particolare è stato dimostrato che un grafo costruito mediante gli archi e i vertici della triangolazione di Delaunay rappresenta un *geometric spanner-tree* [14]: ricordiamo infatti che l'obiettivo di Delaunay è quello di massimizzare gli angoli minimi e quindi, ricordando la legge dei coseni e il teorema generalizzato di Pitagora, si tende a limitare la lunghezza dei lati dei triangoli. È stato mostrato che un cammino minimo calcolato all'interno di un grafo di Delaunay non sarà mai più grande di 2,42 volte un cammino minimo calcolato mediante un grafo completo Euclideo [14]. Più precisamente quest'ultimo può essere definito come un grafo dove l'insieme dei vertici  $V$  è l'insieme  $S$  dei punti presenti nel piano mentre l'insieme degli archi  $E$  è composto da tutti gli archi  $e = (p, q) \quad \forall p, q \in V$ . Inoltre sia  $d(p, q)$  la distanza Euclidea fra i punti  $p$  e  $q$  allora si definirà la funzione peso  $W : E \rightarrow \mathbb{R}$  come la funzione che associa ad ogni arco  $(p, q)$  un peso  $w$  pari alla distanza Euclidea  $d(p, q)$  fra i punti  $p$  e  $q$ .

Date le caratteristiche dei due grafi costruiti, ci aspettiamo che in assenza di vento, il grafo derivato da Voronoi spenda più energia perché due centraline sono in genere più distanti nel grafo Voronoi-oriented che in quello Delaunay-oriented. Tuttavia non è evidente come la direzione degli archi potrà influenzare l'impatto del vento.

È infine necessario sottolineare che si sarebbe potuto usare direttamente il grafo Euclideo completo per descrivere la mappa di routing. Tuttavia sono stati preferiti altri approcci perché nel primo caso sarebbe stato più complesso individuare i venti di riferimento per ciascuna tratta poiché ciascun segmento costruito fra due centraline avrebbe potuto attraversare ben più di due celle di Voronoi e per tratti di diversa lunghezza.

### 3.4.5 Grafo ibrido

Le due configurazioni di grafo precedentemente descritte sono una più densa dell'altra; una con archi più corti dell'altra. Abbiamo quindi proposto una terza configurazione che include tutti gli archi e i vertici delle due precedenti. Questo nuovo grafo è stato denominato *ibrido* perché la sua natura è ibrida. Esso è definito da un insieme di archi e di vertici che sono il risultato dell'unione insiemistica fra gli insiemi del primo e del secondo modello. Il modello è costruito a partire dagli archi ed dai vertici che compongono il grafo Voronoi-Oriented e su di questo vengono aggiunti gli archi di Delaunay che attraversano solo ed esclusivamente due celle di Voronoi. In questo modo si garantisce l'importante proprietà per cui tutti gli archi del grafo hanno una condizione di vento e un conseguente consumo energetico ben definito, ma si rende il grafo più denso, e si cresce il numero di possibili instradamenti del drone. Questi vantaggi nel concreto portano ad avere la possibilità di intraprendere missioni che in caso di condizioni di vento particolarmente avverse prevedano un de-tour che non aumenti eccessivamente sia la durata che la lunghezza del percorso, contenendo i consumi energetici. L'obiettivo del nuovo grafo è aumentare il numero di missioni ammissibili in una data regione.

# Capitolo 4

## Algoritmi per MFP

In questo capitolo discuteremo tre algoritmi che sono proposti per studiare l'ammisibilità di una missione. Innanzitutto, fissata una sorgente (drone warehouse) e un budget, scremiamo le destinazioni che possono essere raggiunte qualsiasi siano le condizioni del vento e quelle che non possono essere raggiunte neppure se le condizioni di vento sono favorevoli su ogni tratto del percorso. Questa scrematura è fatta nella fase di Pre-Processing. Ci concentriamo poi sulle destinazioni raggiungibili a seconda delle condizioni di vento e per tali destinazioni proponiamo tre algoritmi: Offline Shortest Path (OSP), Dynamic Shortest Path (DSP) e Greedy Shortest Path (GSP).

### 4.1 Pre-processing

La procedura di *Pre-Processing* viene utilizzata per ridurre l'insieme dei vertici su cui effettuare i calcoli. In particolar modo ha lo scopo di classificare i vertici di un grafo, data una sorgente e un budget in:

- **GREEN:** un nodo per cui la missione **si può** completare anche assumendo che il consumo energetico di ogni arco sia sempre **massimo**. Per un nodo GREEN, il cammino calcolato durante il Pre-Processing sarà eseguibile qualunque siano le condizioni di vento.
- **BLACK:** un nodo per cui la missione **non si può** completare anche assumendo che il consumo energetico di ogni arco sia sempre **minimo**. Rappresenta la

situazione opposta a quella di un nodo etichettato GREEN: non vi è alcuna speranza di raggiungerlo e quindi viene ignorato.

- **GRAY:** questa risultato è associato a tutti quei vertici per cui la missione non è garantita a priori (cioè in caso di condizioni sempre avverse non si potrà raggiungere), ma è possibile raggiungerlo (sicuramente lo si raggiunge in caso di condizioni favorevoli).

La classificazione così come descritta va a selezionare ed etichettare tutti i nodi con un colore, il cui significato stabilisce se quel nodo è raggiungibile o meno a priori. Pertanto, escludendo i nodi etichettati con GREEN e BLACK la cui fattibilità o non-fattibilità della missione è nota a priori, per eseguire le operazioni dei successivi tre algoritmi verranno utilizzati i soli nodi GRAY. Il fatto di ridursi ai soli nodi BLACK è logica conseguenza, in quanto sono gli unici nodi la cui variazione dei venti sullo scenario può influire in modo significativo sulla fattibilità della missione.

L'algoritmo di Pre-Processing, abbreviato anche come PP, dato il costo energetico minimo e massimo degli archi rispettivamente  $\varepsilon_m = \min \mu(e)$  e  $\varepsilon_M = \max \mu(e)$  determina il grafo di costo minimo  $G^L = (V, E^L)$  dove  $\forall e \in E^L, d(e) = \varepsilon_m \lambda(e)$ . Simmetricamente, si ha il grafo di costo massimo  $G^U = (V, E^U)$  dove  $\forall e \in E^U, d(e) = \varepsilon_M \lambda(e)$ . Per entrambi i grafi è possibile stabilire il consumo energetico unitario minimo e massimo perché considerando il modello fisico di riferimento, avendo discretizzato le possibili velocità e direzioni del vento è possibile estrapolare i parametri che minimizzano o massimizzano questo valore; si va a calcolare il consumo energetico unitario per ogni coppia di valore direzione e velocità del vento possibile, fissato un certo carico e velocità del drone. Una volta calcolate le due matrici  $G^L$  e  $G^U$ , si va a calcolare, fissata la sorgente o vertice di partenza, lo shortest-path  $\pi(v_0, v*)$  e  $\pi(v*, v_0)$  etichettando il vertice di destinazione come BLACK se il consumo minimo è maggiore del budget energetico, GREEN se il consumo massimo è minore del budget a disposizione o GRAY se il budget energetico è sufficiente per soddisfare il consumo minimo ma non per soddisfare quello massimo.

---

**Algorithm 1** PP( $G, B$ )

---

```
1:  $G^L \leftarrow \text{LOWERBOUNDGRAPH}(G, \varepsilon_m)$ 
2:  $G^U \leftarrow \text{LOWERBOUNDGRAPH}(G, \varepsilon_M)$ 
3: for all  $v \in V \setminus v_0$  do
4:    $v.col \leftarrow \text{GRAY}$ 
5:    $C \leftarrow \text{SHORTESTCYCLE}(G^U, v_0, v)$ 
6:   if  $d(C) < B$  then
7:      $v.col \leftarrow \text{GREEN}$ 
8:   end if
9:    $C \leftarrow \text{SHORTESTCYCLE}(G^L, v_0, v)$ 
10:  if  $d(C) > B$  then
11:     $v.col \leftarrow \text{BLACK}$ 
12:  end if
13: end for
```

---

## 4.2 Algoritmi Proposti per la consegna

Gli algoritmi proposti possono terminare in uno dei possibili stati riportati di seguito:

- **CANCELED (C):** questo messaggio di output viene restituito ogni qualvolta la missione viene annullata dal principio, in quanto considerando i consumi energetici valutati in relazione delle condizioni climatiche alla partenza, non si è in grado di portare a termine la totalità della missione.
- **DELIVERED (D):** questo messaggio viene restituito quando il drone intrapresa la missione riesce a giungere alla destinazione ma non è in grado di tornare indietro alla warehouse. Questo tipo di output si ottiene in corrispondenza di un "mezzo successo".
- **FAIL (F):** questo messaggio viene restituito in corrispondenza di una missione fallita prima di raggiungere la prima destinazione. Questo accade ogni volta che la previsione di consumo iniziale definisce una missione fattibile, ma nel corso della stessa i cambiamenti climatici non permettono di completarla (e neppure di raggiungere la destinazione con il carico) a causa di un aumento dei consumi.
- **SUCCESS (S):** questo messaggio viene restituito quando una missione viene completata con successo. Questo accade quando la stima iniziale dei consumi è ammissibile e una volta intrapresa, la missione, nonostante le diverse condizioni di vento, viene completata. In altre parole, la batteria è sufficiente.

### 4.3 Algoritmo Static-Offline Shortest Path

L'algoritmo Offline Shortest Path (OSP), descritto in Algorithm 2, calcola il cammino da percorrere come il ciclo di costo minimo  $C$  ottenuto considerando la funzione costo  $d$  della rilevazione di vento effettuata alla partenza, diciamo  $G_0$ .

---

#### Algorithm 2 OSP( $G, B$ , destination $v$ )

---

```

1:  $S \leftarrow \emptyset$ ,  $flag \leftarrow \text{false}$ ,  $t \leftarrow 0$ 
2:  $C \leftarrow \text{SHORTESTCYCLE}(G_0, v_0, v)$ 
3: if  $d_0(C) \leq B$  then
4:   for all  $e = (v_i, v_j) \in C$  do
5:      $B \leftarrow B - d_t(e)$ 
6:     if  $v_j = v$  then
7:        $flag \leftarrow \text{true}$ 
8:     end if
9:     if  $v_j = v_0$  then
10:       $S \leftarrow \text{SUCCESS}$ 
11:    end if
12:    if  $B < 0$  then
13:      if  $flag = \text{true}$  then
14:         $S \leftarrow \text{DELIVERED}$ 
15:      else
16:         $S \leftarrow \text{FAIL}$ 
17:      end if
18:    end if
19:     $t \leftarrow t + \text{time}(v_i, v_j)$ 
20:  end for
21: else
22:    $S \leftarrow \text{CANCELED}$ 
23: end if
```

---

L'esecuzione parte assegnando allo stato  $S$  il valore nullo e il flag associato alle consegna ultimata valorizzato a *false* (Line 1). Dato il grafo  $G_0$ , il budget energetico  $B$  e la destinazione  $v$ , OSP calcola il ciclo di costo minimo considerando i consumi associati alle condizioni di vento dell'istante iniziale. Quindi non fa altro che calcolare il cammino minimo una sola volta considerando le sole condizioni iniziali. Logicamente se il carica a disposizione è sufficiente per soddisfare la richiesta energetica per il ciclo  $C(d(C) \leq B)$ , allora il drone comincia la sua missione seguendo il percorso calcolato. Tuttavia durante il corso della missione il costo associato agli archi potrebbe variare, in corrispondenza di una variazione di vento, rispetto allo stato iniziale. Proprio per questo si distinguono gli stati *SUCCESS*, *FAIL* e *DELIVERED*. L'algoritmo in questione prevede anche un ulteriore condizione, ovvero se il consumo stimato sotto le condizioni di vento  $G_0$  eccede quello disponibile, allora la missione viene annullata e si restituisce come stato finale *CANCELED*.

Questo algoritmo, data una destinazione calcola il ciclo di costo minimo  $C$  sfruttando l'algoritmo di Dijkstra per calcolare il viaggio di andata e successivamente il viaggio di ritorno. Perciò, l'algoritmo OSP ha una complessità temporale nel caso pessimo (ossia quando per raggiungere la destinazione si esplorino un numero di nodi proporzionale a  $|V|$ ) di  $\mathcal{O}(|E| + |V| \log |V|)$ , considerando un'implementazione di Dijkstra che integra gli Heap di Fibonacci come coda di priorità. Si noti che solo due esecuzioni di Dijkstra sono richieste: una dalla warehouse alla destinazione, ed una dalla destinazione alla warehouse. Ad ogni nodo intermedio, il drone riceve dalla centralina del vertice che ha raggiunto il costo dell'arco che aveva previsto di attraversare. Se il costo è compatibile con il budget procede. L'algoritmo continua fino a che termina o in uno stato di successo o perché il budget diventa negativo e quindi l'arco non è percorribile.

Si noti che OSP richiede uno scambio di dati molto limitato con la centralina.

## 4.4 Algoritmo Dynamic Shortest Path

L'algoritmo Dynamic Shortest Path (DSP) è un algoritmo *online* che calcola dinamicamente il cammino minimo da percorrere ad ogni vertice intermedio che si raggiunge. Questo algoritmo a differenza del precedente non stima una previsione del costo totale del ciclo  $C$ , in quanto a prescindere dalle condizioni ventose iniziali intraprende la missione.

L'algoritmo DSP (riportato in Algorithm 3) funziona nel seguente modo. Inizialmente le variabili vengono tutte inizializzate rispettivamente al loro stato di partenza, rispettivamente  $S, C = \emptyset, t = 0$ . Successivamente si entra all'interno del ciclo principale fintanto che non si terminerà l'esecuzione determinando lo stato finale della missione. La prima operazione svolta è quella di calcolare il cammino minimo dalla sorgente alla destinazione sulla base dei consumi energetici attuali, aggiornando successivamente la batteria disponibile e il nodo su cui il drone è approdato. Al contempo viene anche aggiornata la struttura del grafo rimuovendo l'arco e il vertice appena attraversato. Questo risulta determinante ai fini del completamento della missione, perché evita che il drone entri all'interno di un loop nel quale si muoverà avanti e indietro all'infinito senza poter raggiungere la destinazione, come mostrato nell'Esempio riportato in Figura 4.1. In Figura 4.1, i costi degli archi per la sotto-sequenza

---

**Algorithm 3** DSP( $G, B$ , destination  $v$ )

---

```

1:  $S \leftarrow \emptyset, flag \leftarrow \text{false}, C \leftarrow \emptyset, t \leftarrow 0, u \leftarrow v_0$ 
2: while  $S = \emptyset$  do
3:    $\pi(u, v) \leftarrow \text{SHORTESTPATH}(G_t, u, v)$ 
4:    $e = (u, v_j) \leftarrow \text{primo arco di } \pi(u, v)$ 
5:    $B \leftarrow B - d_t(e), C \leftarrow C \cup e$ 
6:    $V \leftarrow V \setminus u, E \leftarrow E \setminus \text{Adj}(u)$ 
7:    $t \leftarrow t + \text{time}(u, v_j), u \leftarrow v_j$ 
8:   if  $B < 0$  then
9:     if  $flag = \text{true}$  then
10:       $S \leftarrow \text{DELIVERED}$ 
11:    else
12:       $S \leftarrow \text{FAIL}$ 
13:    end if
14:   end if
15:   if  $v_j = v$  then
16:     if  $flag = \text{false}$  then
17:        $flag \leftarrow \text{true}$ 
18:        $(V, E) \leftarrow (G.V, G.E), u \leftarrow v, v \leftarrow v_0$ 
19:     else
20:        $S \leftarrow \text{SUCCESS}$ 
21:     end if
22:   end if
23: end while

```

---

dei grafi "pari"  $G_0, G_2, G_4$  sono raffigurati in (a), mentre i costi per la sotto-sequenza dei grafi "dispari"  $G_1, G_3, G_5$  sono raffigurati in (b). La sorgente del deposito  $s$  è raffigurata in blu, mentre la destinazione  $d$  è raffigurata in rosso. In questo esempio, se non si cancellano gli archi attraversati, il cammino di costo minimo identificato alla partenza è  $s, a, b, d$ . Raggiunto  $a$ , il cammino minimo diventa  $a, s, e, f, d$ . Il drone torna in  $s$ , ma ora il cammino minimo è di nuovo  $s, a, b, d$ . La storia si ripete all'infinito fino ad esaurire l'energia disponibile. Se invece di togliere l'arco verso  $s$  dopo averlo attraversato il cammino verso  $d$  è obbligato. Il ciclo percorso sarà  $s, a, b, d, f, e, s$  con costo complessivo  $1 + 10 + 1 + 10 + 1 + 10 = 33$ . Tale soluzione a seconda del valore di  $B$  potrà condurre a DELIVERED, SUCCESS, o FAIL.

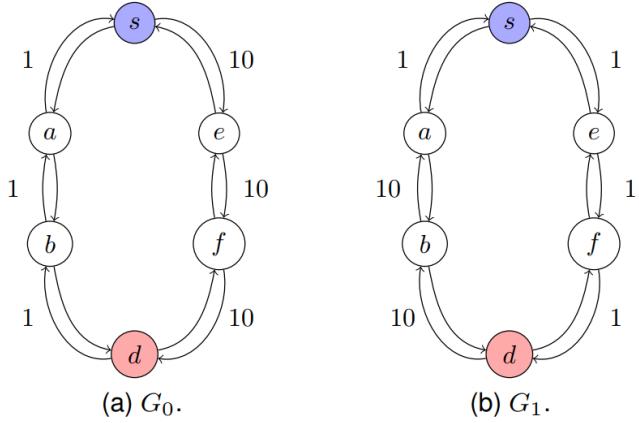


Figura 4.1: Grafo time-dependent che non ammette una soluzione fattibile da  $S$  a  $D$ . Qui, i costi degli archi per la sotto-sequenza dei grafi "pari"  $G_0, G_2, G_4$  sono raffigurati in (a), mentre i costi per la sotto-sequenza dei grafi "dispari"  $G_1, G_3, G_5$  sono raffigurati in (b). La sorgente del deposito  $s$  è raffigurata in blu, mentre la destinazione  $d$  è raffigurata in rosso.

L'analisi della complessità computazionale in tempo, similmente a quanto accaduto per OSP, considerando che si deve ricalcolare ad ogni passo il nuovo shortest-path, richiede  $\mathcal{O}(|C|(|E| + |V| \log |V|))$  operazioni elementari se si considera un'implementazione di Dijkstra che integra gli una coda di priorità efficiente, come gli Heap di Fibonacci, e che termina quando si raggiunge la destinazione (o la source nel secondo tratto). Poiché  $|C|$  è limitato da  $|E|$ , la complessità nel caso pessimo  $\mathcal{O}(|E|^2 + |E||V| \log |V|)$ . Tuttavia, anche in questo caso come in OSP, la complessità è molto minore in generale perché ciascuna esecuzione di Dijkstra non richiede necessariamente di visitare l'intero grafo ma solo i vertici sul ciclo.

Raggiunto un nuovo vertice, la centralina deve passare al drone o il nuovo cammino o tutti i dati perché il drone si calcoli il nuovo cammino. Questo richiede quindi uno scambio di dati fra tutti i vertici.

#### 4.5 Algoritmo Greedy Shortest Path

L'algoritmo Greedy Shortest Path (GSP), riportato di seguito in corrispondenza di Algorithm 4, è un algoritmo greedy che calcola il ciclo  $C$  scegliendo ad ogni vertice raggiunto l'arco uscente non ancora considerato di costo minimo. In particolare, in

ogni istante  $t$ , la strategia applicata è semplicemente quella raggiungere uno dei vertici appartenenti al vicinato, attraversando l'arco di costo minimo considerando il grafo  $G_t$ . Non vi è alcuna garanzia che il GSP calcoli il percorso più breve per raggiungere la destinazione, pur facendo la scelta ottima a livello locale. Similmente al DSP, anche in questo caso si proibisce al drone di attraversare nuovamente vertici e archi già visitati eliminandoli.

---

**Algorithm 4** GSP( $G, B$ , destination  $v$ )

---

```

1:  $S \leftarrow \emptyset$ ,  $flag \leftarrow \text{false}$ ,  $C \leftarrow \emptyset$ ,  $t \leftarrow 0$ ,  $u \leftarrow v_0$ 
2: while  $S = \emptyset$  do
3:    $e = (u, v_j) \leftarrow \arg \min_{j \in Adj(u)} d(u, v_j)$ 
4:   if  $e = \emptyset$  then
5:      $S \leftarrow FAIL$ 
6:   else
7:      $B \leftarrow B - d_t(e)$ ,  $C \leftarrow C \cup e$ 
8:      $V \leftarrow V \setminus u$ ,  $E \leftarrow E \setminus Adj(u)$ 
9:      $t \leftarrow t + time(u, v_j)$ ,  $u \leftarrow v_j$ 
10:    if  $B < 0$  then
11:      if  $flag = \text{true}$  then
12:         $S \leftarrow DELIVERED$ 
13:      else
14:         $S \leftarrow FAIL$ 
15:      end if
16:    end if
17:    if  $v_j = v$  then
18:      if  $flag = \text{true}$  then
19:         $flag \leftarrow \text{true}$ 
20:         $(V, E) \leftarrow (G.V, G.E)$ ,  $u \leftarrow v$ ,  $v \leftarrow v_0$ 
21:      else
22:         $S \leftarrow SUCCESS$ 
23:      end if
24:    end if
25:  end if
26: end while

```

---

La missione viene dichiarata FAIL se il budget residuo non è sufficiente per attraversare ulteriori archi in uscita, o perché il drone non può continuare il suo percorso perché ha esaurito gli archi percorribili, che può verificarsi se il vertice corrente non ha più archi uscenti. È quindi possibile osservare che la differenza tra DSP e GSP è che quest'ultimo seleziona solo il primo arco  $e = (u, v_j)$  il cui costo è minimo rispetto al passo temporale corrente (Linea 3), mentre il primo seleziona in partenza il percorso più breve da  $u$  a  $v$ . Il primo, tuttavia, utilizza solo il primo arco del percorso più breve calcolato. Se non c'è margine disponibile per la visita, la missione viene dichiarata FAIL (Linea 4).

La complessità computazionale in tempo del GSP dell'algoritmo è

$$\mathcal{O}(\max_{v \in V} |Adj(v)| |C|),$$

poiché ad ogni passo dobbiamo ricalcolare l'arco uscente del vertice corrente di costo minimo. Poiché ogni arco può essere visitato una sola volta  $|C| \in \mathcal{O}(|E|)$  e  $\max_{v \in V} |Adj(v)| \in |V|$ . Pertanto la complessità nel caso pessimo è limitata da  $\mathcal{O}(|E||V|)$ . Gli esperimenti hanno poi mostrato che questo algoritmo visita gran parte del grafo, diventando in pratica assai più lento di OSP e DSP.

# Capitolo 5

## Data set e Simulazioni

In questo capitolo si analizzeranno i dati dei venti giornalieri relativi alle differenti aree geografiche della Corsica, forniti direttamente dai pompieri della stessa regione. L'analisi dei dati verrà effettuata seguendo due diversi step: si inizierà esplicitandone la formattazione e le caratteristiche, proseguendo con l'illustrazione di come questi siano stati utilizzati nella fase di simulazione di questo progetto.

Verranno inoltre approfondite le tecniche di costruzione delle diverse tipologie di grafo, introdotte precedentemente, rispetto al caso specifico della Corsica. Infine si esaminerà l'ambiente di testing in cui sono state simulate le differenti missioni che il drone deve svolgere. Da ultimo si mostreranno i risultati ottenuti e saranno analizzati, ad esempio, rispetto al loro impatto economico.

### 5.1 Descrizione dei dati

La Corsica è una regione Francese particolarmente soggetta a incendi boschivi favoriti principalmente dalle caratteristiche morfologiche e climatiche del suo territorio. Poiché il verificarsi di tali fenomeni risulta essere un evento particolarmente dannoso per l'ambiente, si è cercato nel corso del tempo di porre rimedio a queste situazioni problematiche. È stato infatti redatto un piano, in collaborazione con i pompieri del territorio, atto a stabilire i periodi dell'anno e le modalità con cui poter estirpare la vegetazione secca ed altri materiali combustibili attraverso l'accensione di incendi controllati. Per fare ciò si sono basati sulle rilevazioni periodiche delle condizioni ventose locali registrate grazie alle diverse stazioni meteo presenti. La disposizione

di queste stazioni meteo denominate anche con l'appellativo di *centraline* è quella riportata in Figura 5.1.



Figura 5.1: Mappa che descrive la disposizione delle centraline meteo sul suolo della regione Corsica.

È possibile osservare che queste si concentrano principalmente sull'area costiera dell'isola, ma più in generale sull'emisfero settentrionale della stessa (Department of Haute Corse), dove opera l'unità dei pompieri che con cui abbiamo collaborato. Le rilevazioni registrate durante vari periodi dell'anno sono caratterizzate da una particolare formattazione e contengono al loro interno diverse informazioni. Ogni centralina organizza, con un formato comune a tutti, un report delle rilevazioni fatte.

ALISTRO					
num	date	DD	FF	DXI	FXI
20303002	01/02/2020 00:00	230	2,4	220	4,1
20303002	01/02/2020 01:00	220	2,1	230	3,5
20303002	01/02/2020 02:00	220	2,1	210	2,9
20303002	01/02/2020 03:00	200	2,2	200	2,9
20303002	01/02/2020 04:00	210	2,5	190	3,7
20303002	01/02/2020 05:00	220	2	210	3,1
20303002	01/02/2020 06:00	180	0,8	220	2,4
20303002	01/02/2020 07:00	200	1,2	180	1,6
20303002	01/02/2020 08:00	240	1,5	250	1,7
20303002	01/02/2020 09:00	220	1,1	220	2
20303002	01/02/2020 10:00	70	2,5	70	3,6
20303002	01/02/2020 11:00	60	2,7	30	4,2
20303002	01/02/2020 12:00	100	2,1	100	4,1
20303002	01/02/2020 13:00	60	2,7	60	4,5
20303002	01/02/2020 14:00	100	1,7	100	3,8
20303002	01/02/2020 15:00	160	2,3	140	3,6
20303002	01/02/2020 16:00	190	3,2	200	5
20303002	01/02/2020 17:00	200	5,3	200	6,9
20303002	01/02/2020 18:00	200	4,8	210	8,2
20303002	01/02/2020 19:00	190	5	200	6
20303002	01/02/2020 20:00	210	3,2	190	5,8
20303002	01/02/2020 21:00	340	4,3	320	6,7
20303002	01/02/2020 22:00	340	4,1	330	6,3
20303002	01/02/2020 23:00	350	4,5	360	6
20303002	02/02/2020 00:00	340	6,5	340	8,1

Figura 5.2: Esempio di alcune rilevazioni fatte dalla centralina di Alistro.

La tabella riportata nella Figura 5.2 descrive un esempio di alcune rilevazioni fatte dalla stazione meteorologica di Alistro. La tabella appena citata è definita da sei colonne che identificano sei attributi associati ad ogni rilevazione ventosa e tante righe quante sono le rilevazioni registrate. I sei attributi descritti in tabella possono essere così caratterizzati:

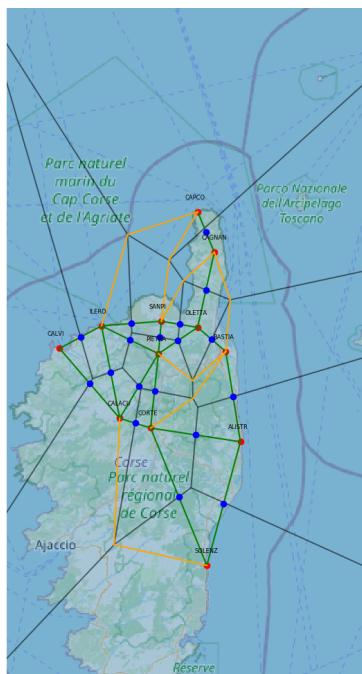
- **num**: questo campo della tabella definisce un identificativo univoco associato alla colonnina che ha effettuatola rilevazione.
- **date**: questo campo così come descritto implicitamente dalla sua etichetta definisce la data e l'ora in cui una certa rilevazione ventosa è stata effettuata. È importante sottolineare che fra una registrazione e l'altra vi è un intervallo di tempo pari a un'ora.
- **DD**: la colonna etichettata come DD contiene l'insieme delle direzioni del vento espresse in gradi rilevate ad un'altitudine di dieci metri.
- **FF**: la colonna etichettata come FF contiene l'insieme delle velocità del vento, espresse in metri al secondo anch'esse ottenute valutando il vento ad un'altitudine di dieci metri.

- **DXI:** il campo DXI definisce la direzione associata alla raffica di vento con maggiore intensità registrata in ogni ora della giornata. Per DXI si tiene in considerazione la stessa altitudine ed unità di misura utilizzata per il campo DD.
- **FXI:** il campo FXI descrive la velocità associata alla raffica di vento con maggiore intensità registrata in ogni ora della giornata. Per FXI si tiene in considerazione la stessa altitudine ed unità di misura utilizzata per il campo FF.

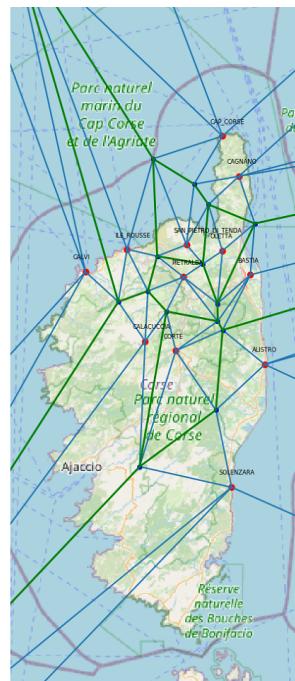
In riferimento al caso di studio preso in esame, le rilevazioni ottenute saranno utilizzate con differenti finalità. Il fine ultimo del progetto è infatti quello di esaminare come i venti influiscano sui consumi energetici relativi agli spostamenti dei droni per le varie consegne. Pertanto, queste rilevazioni verranno utilizzate per simulare missioni di consegna verosimili. Innanzitutto bisogna precisare che il territorio della Corsica è stato suddiviso in celle che rappresentano le aree di influenza di una particolare condizione ventosa e all'interno delle quali vi è esattamente una sola centralina di rilevazione. Durante le simulazioni, fissato un certo orario di partenza, verranno prese in considerazione le condizioni ventose saggiate dalla stessa colonnina. Ad esempio l'area di competenza associata alla centralina di Alistro individuerà un vento la cui combinazione direzione e velocità sarà una di quelle contenute nel dataset.

## 5.2 Costruzione dei grafi

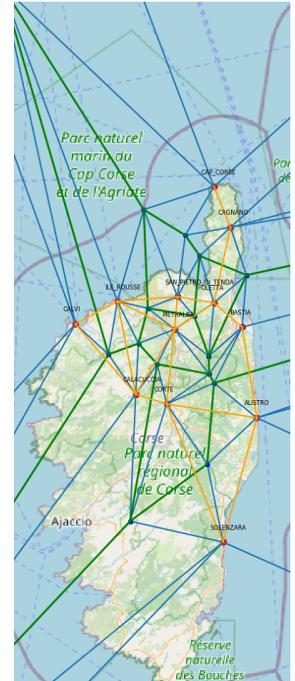
Elemento nevralgico per lo svolgimento della fase di simulazione ed è anche di questa tesi è la costruzione dei tre differenti modelli di grafo relativi allo scenario della Corsica. Nello specifico un ruolo rilevante è ricoperto dalle centraline. Queste rappresentano, riducendo la cartina geografica della Corsica ad un piano bidimensionale, dei punti con coordinate ben definite. Pertanto, prendendo in considerazione i tre modelli di grafo descritti all'interno del Capitolo 3 è possibile costruire delle strutture dati sulla base delle centraline su cui poter simulare lo spostamento dei droni. Così facendo si otterranno le mappe di instradamento specifiche per la Corsica di seguito riportate.



(a) Delaunay-oriented.



(b) Voronoi-oriented.



(c) Ibrido.

Figura 5.3: Differenti grafi.

Questi grafi, come motivato nel Capitolo 3, rappresentano un interessante soluzione per costruire rotte teoriche al fine di studiare il consumo ma soprattutto la fattibilità delle missioni di consegna intraprese dai droni. Tuttavia si era osservato che la strategia più semplice era quella di collegare ogni vertice con tutti gli altri e quindi in uno schema teorico questo poteva essere tradotto come un grafo completo(Figura 5.4) i cui vertici sono rappresentati dalle stazioni meteo sparse sul territorio.

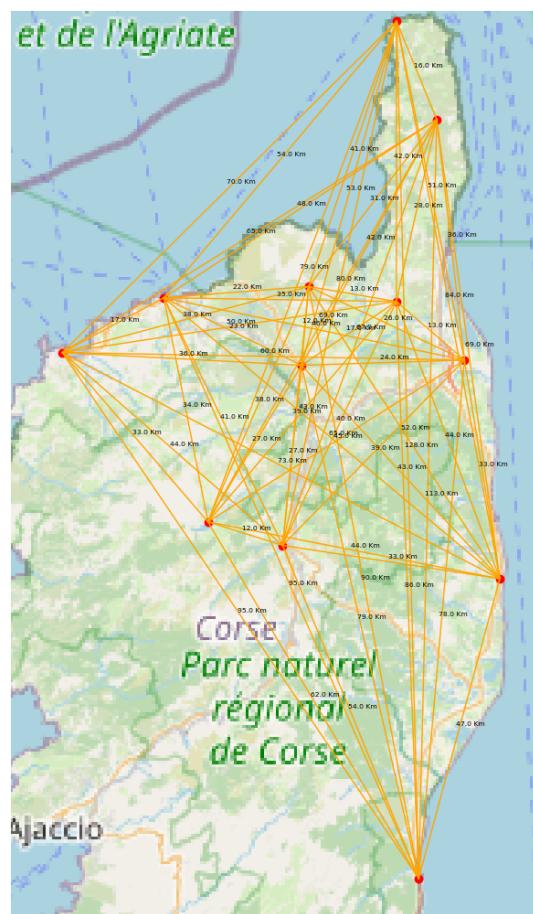


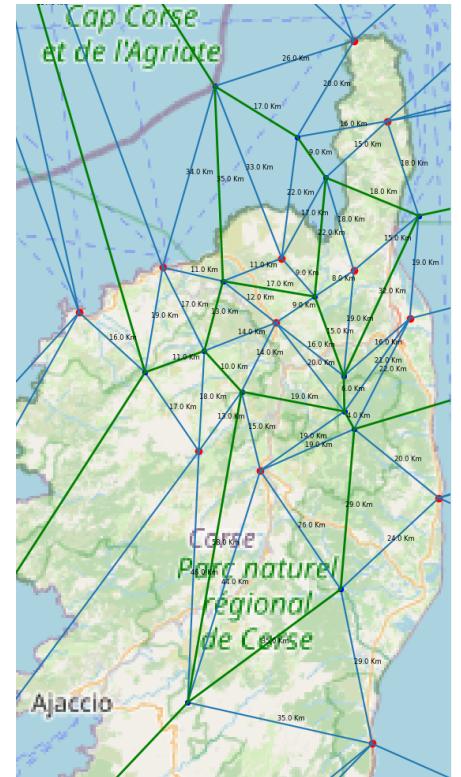
Figura 5.4: Grafo completo costruito considerando come vertici le stazioni meteo della Corsica.

Sebbene il grafo completo sia la configurazione più ovvia per rappresentare lo scenario in cui si sta lavorando, questo complica la questione riguardante la standardizzazione dei venti. Si ha infatti una struttura che non tiene conto dei venti che influenzano determinate tratte, la cartina riportata in Figura 5.4 dimostra questa affermazione, in quanto all'interno di essa vi sono archi che attraversano addirittura

l'intera regione. Contrapposta a questa difficoltà vi è però il vantaggio che gli archi che definiscono questo grafo determinano a loro volta i percorsi più brevi in assoluto per raggiungere da qualsiasi sorgente una qualsiasi destinazione, assumendo però l'assenza di ostacoli. Tuttavia confrontando il grafo completo con i tre modelli costruiti è possibile osservare che le lunghezze dei cammini minimi calcolati all'interno di questi rappresentano un'approssimazione del primo grafo.



(a) Grafo Delaunay-oriented.



(b) Grafo Voronoi-oriented.

Figura 5.5: Grafi pesati con le distanze espresse in chilometri

Calcolando praticamente il rapporto fra le lunghezze dei cammini minimi desunte rispettivamente dal grafo completo e dalle tre configurazioni, si è infatti osservato che queste soluzioni definiscono un'ottima approssimazione del grafo completo. Valutando separatamente i rapporti ottenuti e considerando il massimo rapporto, si ha che il grafo Voronoi-oriented è una 1.203 approssimazione del grafo completo Euclideo, mentre il grafo Delaunay-oriented e Hybrid rappresentano rispettivamente una 1.04 e

1.032 approssimazione. Ovviamente questi rapporti sono solo significativi per questo caso particolare analizzato.

Destinazioni	Voronoi(Km)	Delaunay(Km)	Hybrid(Km)	Completo(Km)
Alistro	53.31	46.58	46.58	46.58
Corte	55.01	53.96	53.56	53.96
Calacuccia	80.54	65.68	65.67	62.11
Calvi	107.38	98.96	98.65	95.09
Ile Rousse	96.70	99.69	95.65	94.86
Pietralba	82.19	80.97	80.97	78.79
Bastia	80.66	79.62	79.62	77.89
San Pietro di Tenda	92.81	92.73	92.73	90.04
Oletta	87.53	92.99	87.52	86.36
Cagnano	117.37	116.33	115.19	113.33
Cap Corse	134.21	132.00	130.86	127.85

Tabella 5.1: Tabella dei cammini minimi calcolati con sorgente in Solenzara.

Quanto riportato nella Tabella. 5.1 consolida la stretta relazione tra i tre grafi ed il grafo completo Euclideo.

Sotto queste assunzioni è possibile quindi passare alla fase di simulazione vera e propria, applicando gli algoritmi descritti in precedenza sui vari scenari costruiti e sotto condizioni di vento variabili nel corso del tempo. Si propone in Figura 5.6 di seguito riportata un esempio di esecuzione per illustrare i differenti stati definiti nel capitolo precedente. Per semplicità si farà riferimento ad un grafo che simula un percorso rettilineo influenzato dai venti costanti.

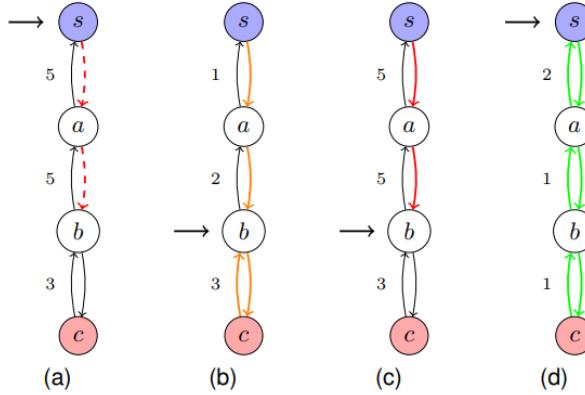


Figura 5.6: Esempi di missioni con  $B = 10$ . La freccia indica dove si trova il drone. gli archi tratteggiati e continui rappresentano, rispettivamente, gli archi pianificati (non percorsi) e quelli percorsi. Il deposito è indicato con il blu e la destinazione con il rosso. In dettaglio: CANCELED (solo OSP) in (a); DELIVERED in (b); FAIL in (c); SUCCESS in (d).

Concludiamo la trattazione di questa sezione, introducendo una breve riflessione sull’attuazione delle simulazioni e di come queste sono in stretto rapporto con la procedura di Pre-Processing. La procedura di Pre-Processing, così come descritto nella Sezione 4.1, è stata sviluppata con lo scopo di individuare, fissata la sorgente e le condizioni dell’ambiente in cui si sta lavorando, i vertici su cui l’esito della missione di consegna non è noto a priori. Come già detto, si distinguono tre diverse tipologie di vertici: GREEN quando la carica è sufficiente per terminare la missione con le condizioni di vento peggiori, BLACK quando la carica non è sufficiente per terminare la missione con le condizioni di vento migliori e GRAY quando il successo o il fallimento non si può determinare a priori. A tal proposito è molto importante osservare che eseguendo la procedura di Pre-Processing sugli scenari appena descritti si hanno dei risultati piuttosto negativi, in quanto come mostrato nella Tabella 5.2, posizionando il deposito a Solenzara tutte le destinazioni possibili risultano essere non raggiungibili anche con le condizioni ventose ottimali.

Questo risultato, seppur ampiamente negativo non è sorprendente, in considerazione del fatto che i modelli di octocopter attualmente disponibili sul mercato non sono in grado di coprire distanze così grandi. Tali distanze infatti risultano essere proibitive principalmente perché le capacità delle batterie attualmente impiegate non sono in grado di supportare un tale sforzo. Pertanto, al fine di descrivere simulazioni

Destinazioni	VoronoiO	DelaunayO	Hybrid
Alistro	BLACK	BLACK	BLACK
Corte	BLACK	BLACK	BLACK
Calacuccia	BLACK	BLACK	BLACK
Calvi	BLACK	BLACK	BLACK
Ile Rousse	BLACK	BLACK	BLACK
Pietralba	BLACK	BLACK	BLACK
Bastia	BLACK	BLACK	BLACK
San Pietro di Tenda	BLACK	BLACK	BLACK
Oletta	BLACK	BLACK	BLACK
Cagnano	BLACK	BLACK	BLACK
Cap Corse	BLACK	BLACK	BLACK

Tabella 5.2: Tabella dei risultati del Pre-Processing con sorgente in Solenzara e batteria con 5000 kJ per i differenti modelli di grafo.

significative si è deciso di scalare le distanze degli archi riducendole di un ordine di grandezza. La dimensione della riduzione è dovuta dal fatto che considerando una batteria di 5000 kJ i consumi valutati con le distanze reali risultano essere dieci volte superiori a quelli spendibili. Così facendo, utilizzando una comune batteria da 5000 kJ si è in grado di raggiungere tutti i punti della mappa con condizioni di vento favorevoli. Tuttavia riducendo le distanze si riducono anche i tempi di percorrenza degli archi e per questa ragione si riduce drasticamente anche il numero di variazioni di vento che si incontrano durante la missione. Questo fenomeno accade in considerazione del fatto che la differenza di tempo che c'è fra una registrazione e l'altra è di un'ora, ma avendo ridotto le distanze il drone è in grado di completare gran parte delle missioni con meno di un'ora di viaggio. Per questo motivo è stata applicata una riduzione anche agli intervalli di tempo con cui il vento varia al fine di avere perturbazioni con la stessa frequenza che si avrebbe in condizioni "normali".

### 5.2.1 Scalatura dei dati

Per superare i risultati negativi appena visti, vi sono due vie percorribili: 1) aumentare la capacità della batteria, o 2) ridurre le distanze degli archi. La scelta fra le due possibili alternative è ricaduta sulla riduzione delle distanze per costruire un esempio quasi-reale per le batterie di un comune octocopter. In particolare si è deciso di

scalare le distanze di dieci volte rispetto ai valori reali, proprio in considerazione dei risultati ottenuti durante l'esecuzione del Pre-Processing. Come già detto oltre a scalare le distanze, si è scalata la frequenza con cui le condizioni ventose si modificano, passando da cambiamenti ogni ora a cambiamenti ogni quarto di ora. Per quanto riguarda la riduzione dei tempi di aggiornamento, si è scelto di ridurli di quattro volte anziché di dieci in quanto rappresenta un giusto compromesso fra numero di possibili cambiamenti durante una singola missione e "realisticità" della simulazione.

I dati così corretti sono riferiti nel resto del lavoro come dati quasi-reali.

## 5.3 Simulazioni

In questa sezione verranno descritte le specifiche tecniche utilizzate per automatizzare le simulazioni sui tre diversi grafi costruiti.

### 5.3.1 Test su dati 'quasi reali'

Dopo aver descritto come i dati sono stati trattati e modellati per la fase di simulazione, illustriamo i tecnicismi e le caratteristiche della procedura che la implementa. Lo pseudo-codice riportato di seguito definisce il cuore della fase di simulazione, in quanto sintetizza i passi salienti del codice prodotto.

---

#### Algorithm 5 nTest( $G, B, H$ )

---

```

1:  $dictResultOSP \leftarrow \emptyset$ ,  $dictResultDSP \leftarrow \emptyset$ ,  $dictResultGSP \leftarrow \emptyset$ 
2: for all  $s \in V$  do
3:    $listGray \leftarrow PP(G, B, s)$ 
4:   for all  $d \in listGray$  do
5:     for all  $hour \in H$  do
6:        $dictResultOSP \leftarrow dictResultOSP + OSP(G, B, s, d, hour)$ 
7:        $dictResultDSP \leftarrow dictResultDSP + DSP(G, B, s, d, hour)$ 
8:        $dictResultGSP \leftarrow dictResultGSP + GDP(G, B, s, d, hour)$ 
9:     end for
10:     $save(dictResultOSP), save(dictResultDSP), save(dictResultGSP)$ 
11:     $dictResultOSP \leftarrow \emptyset, dictResultDSP \leftarrow \emptyset, dictResultGSP \leftarrow \emptyset$ 
12:  end for
13: end for

```

---

L'algoritmo  $nTest$ , riportato in corrispondenza di Algorithm 5, svolge la simulazione delle varie missioni considerando i soli nodi grigi restituiti dalla procedura di Pre-Processing (Linea 3). Una volta inizializzati i dizionari utilizzati per il calcolo del

numero di successi e fallimenti riscontrati (Linea 1), si innestano tre cicli che scandiscono i vertici e gli istanti di tempo  $H$  su cui calcolare la simulazione. Il primo ciclo scorre tutte le possibili sorgenti all'interno dell'insieme dei vertici  $V$ , dopodiché richiamando la funzione PP calcola quali nodi a partire dalla sorgente fissata sono GREEN, BLACK o GRAY; per semplicità nello pseudo-codice si è ristretto l'output di PP ai soli nodi GRAY. A questo punto si scorrono attraverso un altro ciclo le destinazioni, i nodi grigi ottenuti, su cui simulare la consegna (Linea 4). Infine si avvia l'esecuzione dei tre algoritmi richiamando le rispettive procedure che li calcolano, innestandoli all'interno di un ulteriore ciclo che scandisce i diversi orari di partenza della consegna. L'output dei tre algoritmi è un dizionario le cui chiavi sono gli stati mentre i valori a loro associati definiscono il numero di occorrenze ottenute. Completata l'esecuzione per una certa sorgente e destinazione, vengono salvati i risultati (Linea 10) e subito di seguito resettati i tre dizionari (Linea 11).

La complessità della procedura nTest, indicando le complessità osservate per le diverse procedure richiamate al suo interno semplicemente con  $T_{\text{alg}}$ , è  $\mathcal{O}(|V||\mathcal{G}||\mathcal{H}|T_{\text{alg}})$ , dove  $\mathcal{G}$  è il numero dei nodi grigi e  $|\mathcal{H}|$  è la cardinalità dell'insieme degli orari su cui è stata fatta la simulazione. Pertanto la complessità nel caso pessimo è limitata da  $\mathcal{O}(|V|^2|T_{\text{alg}}|)$  perché  $|\mathcal{H}| = 30$  è una costante ed è omesso dalla complessità asintotica. Questa analisi teorica della complessità rivela che la procedura di simulazione risulta essere particolarmente pesante dal punto delle computazioni. Infatti ciascun algoritmo viene chiamato per ogni coppia di vertici ed ogni coppia è esaminata  $|\mathcal{H}|$ . La ripetizione dell'algoritmo  $|\mathcal{H}|$  volte è richiesta perché l'output degli algoritmi dipende dal vento e dall'orario e quindi ripetendo l'algoritmo possiamo restituire il comportamento medio degli algoritmi. Diciamo che “stabilizziamo” i risultati.

## 5.4 Risultati

In questa sezione si presentano i risultati estrapolati dalle differenti simulazioni eseguite. I primi valori che passeremo in rassegna sono quelli calcolati dalla funzione di Pre-Processing. In particolare è stato calcolato il numero di vertici GREEN, BLACK e GRAY sotto diverse condizioni di batteria, carico e scenario.

Dagli histogrammi sotto riportati è possibile osservare innanzitutto che la scalatura applicata ha avuto effetti positivi per la fattibilità delle missioni, in quanto

con una batteria completamente carica non si hanno nodi non raggiungibili a priori anche considerando il carico all'andata di 7 kg. Scendendo nel dettaglio è possibile osservare che disponendo del budget energetico massimo i grafi Delaunay-oriented (DG) e ibrido (HG) sono quelli che hanno un maggior numero di nodi verdi a scapito del grafo Voronoi-oriented (VG) in cui seppur di poco si ha un maggior numero di vertici grigi. Se si valutano i risultati del Pre-Processing con batteria carica a metà spicca la presenza di nodi neri e un numero più che dimezzato di nodi verdi. Sotto questa ipotesi il grafo ibrido è quello che ha il minor numero di nodi neri, e quindi rappresenta la soluzione che potenzialmente può garantire la massima fattibilità delle missioni. È anche importante notare l'impatto del  $m_p$  nella differenza di budget: con un carico di 7 kg e mezza batteria i risultati peggiorano significativamente.

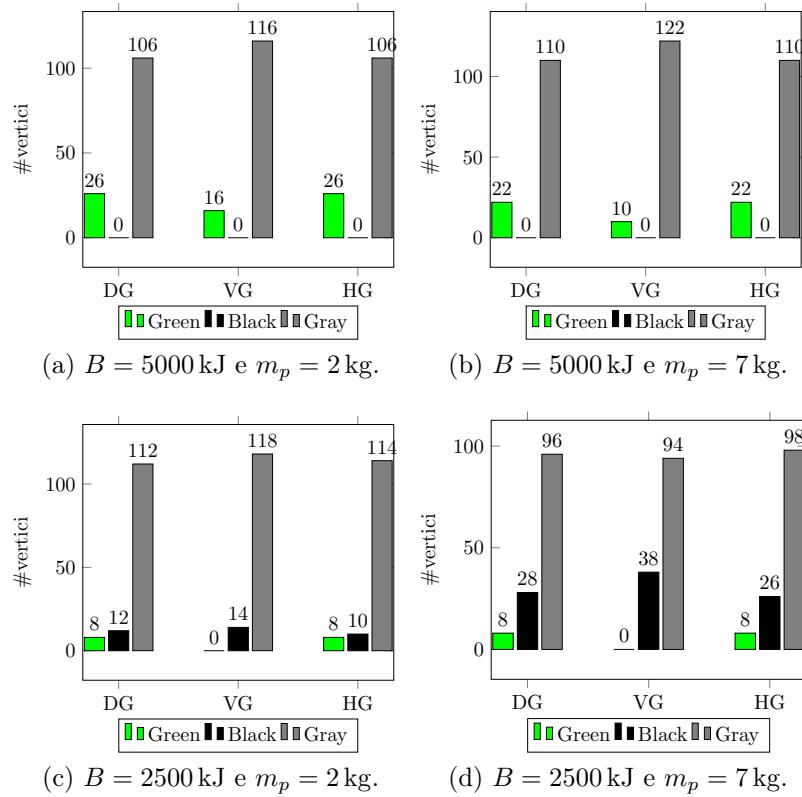


Figura 5.7: Risultati Pre-Processing rispetto al numero complessivo di vertici

Ora si continua confrontando le performance dei tre algoritmi valutandone gli stati restituiti. In particolare si valuterà la percentuale di SUCCESS, FAIL, DELIVERED e CANCELED (solo per OSP) ottenuta da trenta simulazioni, dove ognuna di esse avrà

un differente orario di partenza. Questo è stato svolto per tutte le possibili località di partenza considerando i soli vertici grigi restituiti dal Pre-Processing. Osservando i valori riportati nelle tabelle da una macro-prospettiva si può notare che sia DSP che OSP risultano essere particolarmente efficienti con un numero molto elevato di successi e di consegne senza ritorno, mentre l'algoritmo GSP si dimostra nettamente inferiore ai precedenti e nella sua globalità non è molto efficace per portare a termine le missioni. Quanto osservato è il trend evidenziato nella totalità delle simulazioni.

Scendendo nel dettaglio, se si paragonano i risultati ottenuti per DSP e OSP abbiamo che nel complesso il primo ha un numero di SUCCESS di poche unità superiori rispetto al secondo. Tuttavia il valore che si discosta maggiormente è quello degli stati DELIVERED in cui DSP è nettamente in vantaggio rispetto ad OSP. Il divario nei DELIVERED si verifica in virtù della natura dinamica di DSP e statica di OSP in quanto il primo algoritmo come già illustrato fa intraprendere la missione al drone a priori senza fare una stima del consumo energetico richiesto per affrontare la consegna. Questo non accade per OSP il quale calcola una proiezione del consumo energetico associata al ciclo di costo minimo con le condizioni ventose di partenza. Pertanto, OSP perde distanze di DELIVERED da DSP perché in caso di condizioni ventose iniziali particolarmente avverse decide di annullare la missione dal principio. Quindi DSP ha un numero più alto di DELIVERED e in parte di SUCCESS perché in corrispondenza di un CANCELED per OSP, esso intraprendere la missione completandola parzialmente o totalmente. Questo andamento si fa molto più marcato in corrispondenza dei test effettuati con budget energetico dimezzato e con otto classi di vento, poiché rendono molto più rilevanti le condizioni di vento che pervadono lo scenario.

Discorso totalmente diverso deve essere fatto per l'algoritmo GSP, in quanto le sue prestazioni particolarmente negative vanno imputate al suo modo di scegliere gli archi un po' troppo goloso. L'algoritmo GSP infatti seleziona ad ogni passo l'arco che localmente è meno costoso in termini di consumi energetici senza però valutare il percorso globale che sta intraprendendo. Questo tradotto nella pratica potrebbe voler dire che la scelta ottima locale potrebbe portarmi lontano dalla destinazione e quindi lontano dalla scelta ottima globale. La costatazione appena fatta oltre ad essere la motivazione dell'alto numero di fallimenti è anche la motivazione dei lunghi tempi richiesti da GSP per convergere alla soluzione. Intrapresa la missione, l'algo-

ritmo nella maggior parte dei suoi fallimenti porta o all'esaurimento della batteria o all'esaurimento degli archi percorribili causando in questo modo un aumento dei tempi di esecuzione.

Tuttavia se si confrontano i tre algoritmi dal punto di vista della conoscenza richiesta per il loro funzionamento si ha un inversione dei ruoli. Il funzionamento alla base dell'algoritmo GSP richiede infatti una quantità minore di informazioni sullo scenario, in quanto per individuare l'arco che si dovrà percorrere richiede la conoscenza dei soli consumi energetici associati agli archi uscenti relativi alla sua posizione corrente. In questa ipotetica classifica segue l'algoritmo OSP, in quanto per poter selezionare il cammino da dover percorrere ha la necessità di una conoscenza completa dei consumi energetici iniziali, ma una conoscenza nettamente ridotta nei passi che seguono. Si ha infatti che una volta calcolato il percorso da seguire le sole informazioni richieste sono i consumi associati agli archi del percorso modulati sulla base delle nuove condizioni di vento. Questo colloca ancora una volta l'algoritmo OSP in una posizione intermedia. Infine conclude DSP in relazione al fatto che per la scelta degli archi da attraversare ha la necessità di una conoscenza globale dei consumi energetici. Questo perché DSP ad ogni passo ricalcola il cammino minimo dal vertice corrente verso la destinazione richiedendo appunto una costante conoscenza globale dell'ambiente in cui si sta lavorando.

Nel seguito in Figura 5.8 e Figura 5.9 riportiamo in dettaglio per ogni tipo di grafo, DG, VG e HG, i risultati ottenuti. Fissato un algoritmo, fissato un budget energetico, e fissato un numero di classi di vento, plottiamo per ogni sorgente la percentuale di stati rilevati per ciascun algoritmo. In particolare, fissata la sorgente, plottiamo la percentuale di SUCCESS, DELIVERED, FAIL, e CANCELED per ciascuno dei tre algoritmi proposti. Si ricorda che lo stato CANCELED è ritornato solo da OSP. L'algoritmo OSP, con budget massimo e 16 classi di vento, per quasi tutte le sorgenti, raggiunge con successo tutte le altre destinazioni. Lo stesso per DSP. Invece GSP, ha risultati molto scarsi.

Scendendo da 16 a 8 classi di vento, i risultati di OSP peggiorano leggermente. Quindi un maggior dettaglio nei consumi, aiuta OSP: sono più gli archi in cui guadagna raffinando il vento che quelli in cui perde. Tutti gli algoritmi migliorano le loro prestazioni passando da 8 a 16 classi di vento. I guadagni maggiori aumentando le classi dei venti si hanno sul grafo VG probabilmente perchè gli archi sono più lunghi.

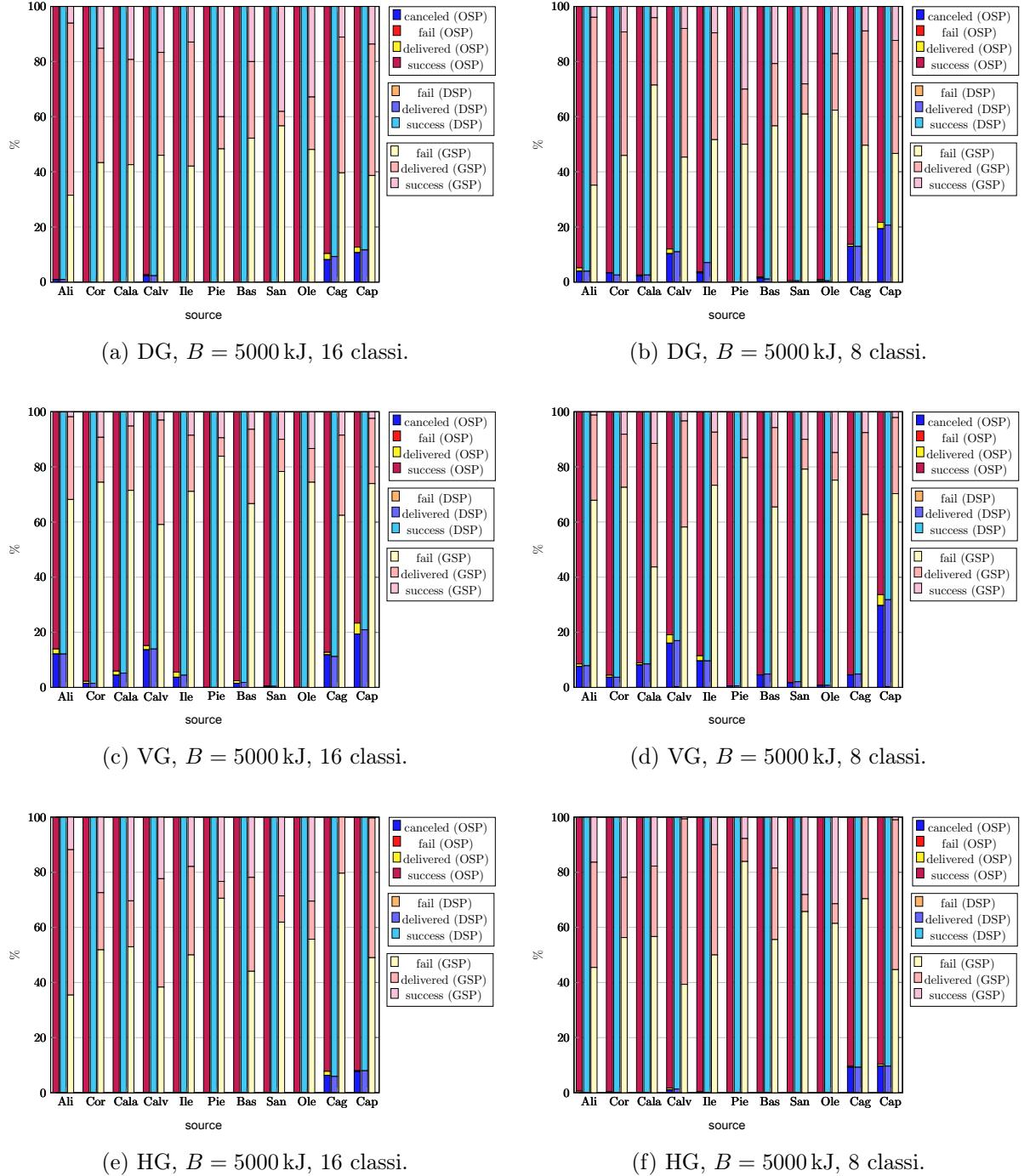


Figura 5.8: Risultati delle simulazioni che descrivono in percentuale gli stati ottenuti fissato  $B = 5000$  kJ.

In Figura 5.9, si riportano i risultati con  $B = 2500$  kJ. I plot appaiono molto più frammentati. Come prima OSP e DSP sono molto simili, ma è evidente ora che DSP recupera come DELIVERED le missioni che OSP ritorna come CANCELED. Dimezzando la batteria, si dimezzano anche i SUCCESS. Come già visto, i risultati leggermente migliorano passando da 8 a 16 classi. I risultati sono leggermente peggiori per VG che ha gli archi più lunghi. D'altra parte su HG gli algoritmi migliorano le prestazioni per la maggior connettività. Ancora una volta, sorgenti diverse si comportano in modo simile.

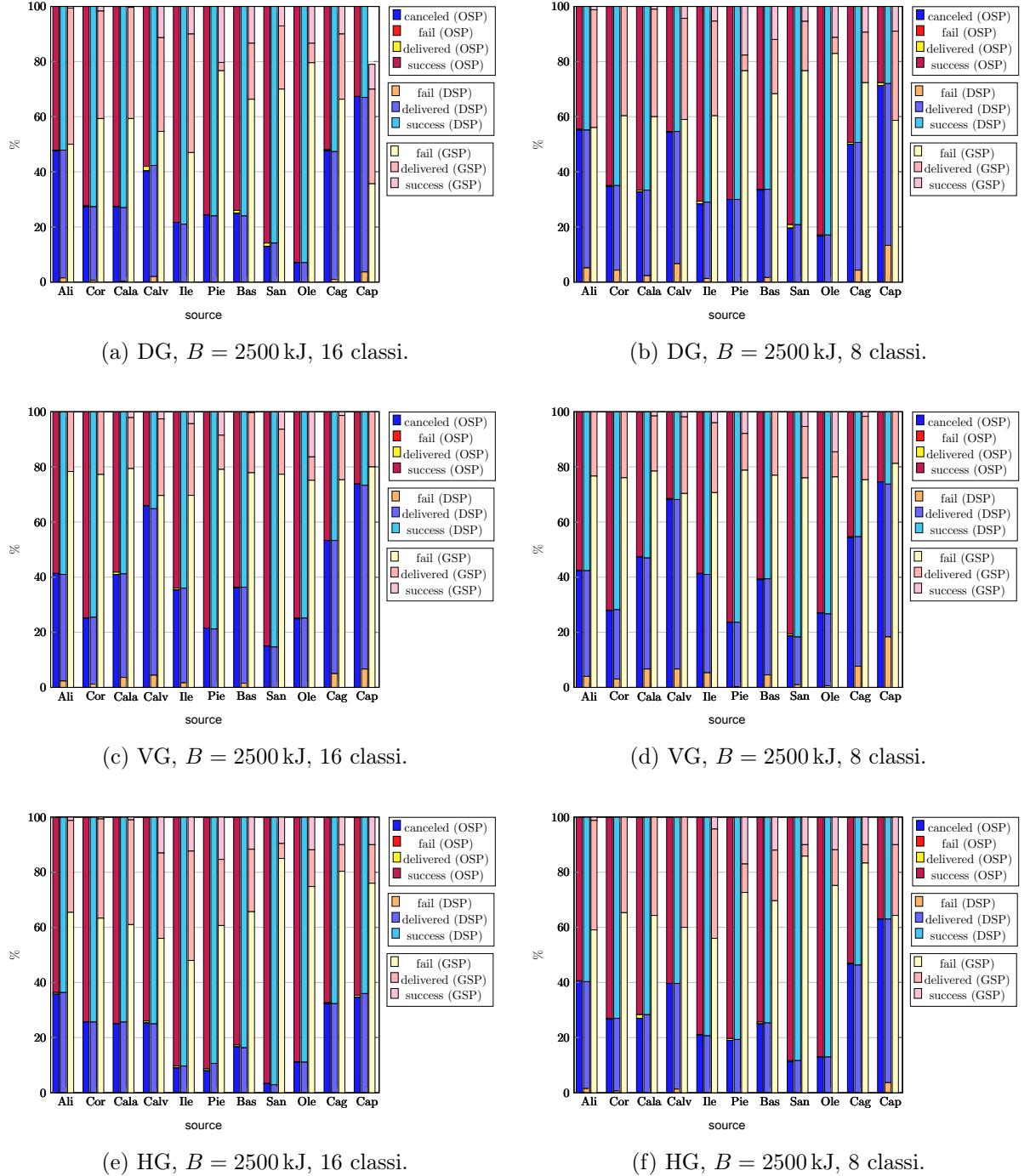


Figura 5.9: Risultati delle simulazioni che descrivono in percentuale gli stati ottenuti fissato  $B = 2500$  kJ.

Interessante è vedere che le sorgenti non giocano un ruolo così importante. Esse si comportano in modo analogo, forse con l'eccezione delle ultime due sorgenti che sono le più estreme (geograficamente). Questo ci permette di sintetizzare i risultati come riportato in Tabella 5.3. Per ogni algoritmo e per ogni tipo di grafo, si calcola il numero medio di outcome su tutte le sorgenti e lo si riporta in percentuale. Questa tabella conferma quanto affermato in precedenza, e permette di confrontare i risultati ottenuti con le diverse classi di vento. Si evidenzia che complessivamente l'utilizzo di più classi di vento non modifica significativamente l'andamento dei risultati. L'andamento al variare delle classi di vento era modificato in modo più significativo per ciascuna sorgente. L'algoritmo che più beneficia del maggior numero di classi è DSP sul grafo Delaunay-oriented. Questo è interessante, ma non sorprendente, perché la triangolazione Delaunay mira a massimizzare l'ampiezza dell'angolo minimo in ogni triangolo costruito. Come già osservato, DSP non migliora il numero di SUCCESS ma solo aumenta il numero di consegne eseguite, ossia DELIVERED. Con il 50% del budget, ossia  $B = 2500 \text{ kJ}$ , OSP cancella più missioni ma non fallisce, mentre al più il 5% delle missioni di DSP fallisce. Un'elevata percentuale di missioni invece fallisce con DSP.

Tabella 5.3: Confronto tra OSP, DSP, e GSP su dati quasi-reali. C, S, D, e F rappresentano CANCELED, SUCCESS, DELIVERED, e FAIL, rispettivamente. Dati sono espressi in percentuale.

B	gray	OSP				DSP			GSP			
		C	S	D	F	S	D	F	S	D	F	
8 wind classes												
VG	100	88	9	90	1	0	90	10	0	7	25	68
	50	89	44	56	0	0	56	39	5	3	21	76
DG	100	80	7	92	1	0	93	7	0	13	36	51
	50	85	42	57	1	0	58	38	4	6	29	64
HG	100	80	3	97	0	0	97	3	0	13	31	56
	50	86	34	66	0	0	66	33	1	6	27	67
16 wind classes												
VG	100	88	8	91	1	0	92	8	0	7	23	71
	50	89	42	58	0	0	58	40	2	3	21	76
DG	100	80	3	96	1	0	97	3	0	19	38	44
	50	85	35	65	1	0	65	34	1	8	32	58
HG	100	80	2	98	0	0	98	2	0	18	30	52
	50	86	25	75	0	0	75	25	0	8	26	66

# Capitolo 6

## Conclusioni e lavori futuri

In questo lavoro, è stato introdotto il MFP che studia la possibilità di intraprendere, mediante l'uso di un drone, consegne di merci a partire da un deposito verso un dato cliente con un budget limitato di batteria. Per modellare correttamente il MFP, sono state proposte diverse tipologie di grafo definite con una funzione peso dipendente dal tempo. Sono stati implementati tre algoritmi, OSP, DSP e GSP, e infine sono state valutate le loro prestazioni in termini di missioni compiute su dati quasi reali. Questa indagine ha quindi rivelato l'importanza di conoscere le condizioni di vento al fine di determinare cammini energeticamente sostenibili da parte dei droni. Questa ipotesi è stata infatti messa in risalto contrapponendo i risultati ottenuti dai tre algoritmi sui differenti scenari costruiti. È stato riscontrato come un approccio molto conservativo, che inizia una missione solo se le condizioni iniziali del vento ci permettono di completarla, come fa OSP, annulla fino al 9% delle consegne se le rotte sono lunghe; mentre si è visto che un approccio più rischioso, come il DSP, seppur richieda maggiori informazioni, è in grado di completare tutte le consegne riportando con successo il drone al deposito in più del 90% delle missioni. D'altra parte, una strategia a breve termine che considera le sole condizioni di vento locali, può fallire più del 50% delle consegne abbandonando il drone da qualche parte all'interno della mappa. Da ciò si evidenzia che nell'effettuare consegne con l'ausilio dei droni, più in particolare con gli octocopter, è molto importante, se non decisivo, pianificare il cammino tenendo in considerazione i venti che influenzano l'ambiente e i consumi energetici dedotti.

Moltissimo rimane da fare in quanto questi sono gli algoritmi più semplici che vengono in mente e sono stati studiati solo per verificare l'ammissibilità di una destina-

zione. Potremmo invece interessarci a soluzioni che ottimizzano il consumo energetico o la durata del percorso o che minimizzano l'energia spesa dal sistema per ricalcolare la rotta. Fra queste ultime, ad esempio, si possono studiare soluzioni pre-calcolate che considerano i cambi di vento più frequenti, evitando il calcolo online eseguito dal DSP. Riguardo l'ottimizzazione del consumo energetico, si potrebbe introdurre la possibilità che durante la missione il drone si fermi ed attenda le condizioni migliori di vento per raggiungere la destinazione. Durante tali attese, il drone potrebbe ricaricarsi in 'piazzole ad-hoc', incrementando il budget a disposizione. Sono interessanti anche soluzioni che considerano punti di ricarica nel percorso ma per ottimizzare i tempi per la consegna. Inoltre, in questo lavoro, le rotte sono predefinite. Potremmo invece definire grafi con rotte dinamiche che sfruttano il vento favorevole al momento.

# Appendice A

## Risultati estesi

Nelle tabelle di seguito sono mostrati gli stati ottenuti utilizzando differenti tipologie di grafo, cioè, Voronoi-oriented, Delaunay-oriented, ed ibrido, variando la capacità della batteria  $B$ , carico del drone  $m_p$ , e per differente numero di classi di vento.

Sorgente	N. GRAY	OSP				DSP				GSP		
		S	F	D	C	S	D	F	S	D	F	
Solenzara	11	8,23	0	0,2	2,57	8,53	2,47	0	0,03	3,7	7,27	
Alistro	11	9,47	0	0,2	1,33	9,67	0,71	0	0,2	3,3	7,5	
Corte	9	8,8	0	0,07	0,13	8,87	0,13	0	0,83	1,47	6,7	
Calacuccia	9	8,47	0	0,13	0,4	8,53	0,47	0	0,47	2,1	6,43	
Calvi	11	9,33	0	0,17	1,5	9,47	1,53	0	0,33	4,17	6,5	
Ile Rousse	9	8,5	0	0,17	0,33	8,6	0,4	0	0,77	1,83	6,42	
Pietralba	6	6	0	0	0	6	0	0	0,57	0,4	5,03	
Bastia	11	10,73	0	0,1	0,17	10,8	0,2	0	0,7	2,97	7,33	
San Pietro Tenda	8	7,97	0	0	0,03	7,97	0,03	0	0,8	0,93	6,27	
Oletta	9	9	0	0	0	9	0	0	1,2	1,1	6,7	
Cagnano	11	9,6	0	0,1	1,3	9,77	1,23	0	0,93	3,2	6,87	
Cap Corse	11	8,43	0	0,43	2,13	8,7	2,3	0	0,27	2,6	8,13	

Tabella A.1: Voronoi-oriented,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP				GSP		
		S	F	D	C	S	D	F	S	D	F	
Solenzara	6	1,87	0	0,07	4,07	1,87	4,13	0	0	1,97	4,03	
Alistro	10	5,87	0	0	4,13	5,9	3,87	0,23	0	2,17	7,83	
Corte	11	8,23	0	0	2,77	8,2	2,67	0,13	0	2,5	8,5	
Calacuccia	11	6,4	0	0,1	4,5	6,47	4,13	0,4	0,23	2,03	8,73	
Calvi	9	3,07	0	0	5,93	3,17	5,43	0,4	0,23	2,5	6,27	
Ile Rousse	10	6,4	0	0,07	3,53	6,4	3,43	0,17	0,43	2,6	6,97	
Pietralba	11	8,63	0	0	2,37	8,67	2,33	0	0,93	1,37	8,7	
Bastia	11	7	0	0,03	3,97	7	3,83	0,17	0,03	2,4	8,57	
San Pietro Tenda	10	8,5	0	0	1,5	8,53	1,47	0	0,63	1,63	7,73	
Oletta	11	8,23	0	0,03	2,73	8,23	2,77	0	1,8	0,93	8,27	
Cagnano	10	4,67	0	0	5,33	4,67	4,83	0,5	0,13	2,33	7,53	
Cap Corse	8	2,1	0	0	5,9	2,13	5,33	0,53	0	1,6	6,4	

Tabella A.2: Voronoi-oriented,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP				GSP		
		S	F	D	C	S	D	F	S	D	F	
Solenzara	11	9,17	0	0,23	1,6	9,27	1,73	0	0,07	7,17	3,77	
Alistro	11	10,9	0	0	0,1	10,9	0,1	0	0,67	6,87	3,47	
Corte	9	9	0	0	0	9	0	0	1,37	3,73	3,9	
Calacuccia	9	9	0	0	0	9	0	0	1,73	3,43	3,83	
Calvi	10	9,73	0	0,03	0,23	9,77	0,23	0	1,67	3,73	4,6	
Ile Rousse	8	8	0	0	0	8	0	0	1,03	3,6	3,37	
Pietralba	6	6	0	0	0	6	0	0	2,4	0,7	2,9	
Bastia	9	9	0	0	0	9	0	0	1,8	2,5	4,7	
San Pietro Tenda	7	7	0	0	0	7	0	0	2,67	0,37	3,97	
Oletta	7	7	0	0	0	7	0	0	2,3	1,33	3,37	
Cagnano	9	8,07	0	0,2	0,73	8,17	0,83	0	1	4,43	3,57	
Cap Corse	10	8,73	0	0,2	1,07	8,83	1,17	0	1,37	4,77	3,87	

Tabella A.3: Delaunay-oriented,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	5	1,5	0	0,07	3,43	1,5	3,5	0	0	3,27	1,73
Alistro	11	5,73	0	0,03	5,23	5,73	5,1	0,17	0,07	5,43	5,5
Corte	10	7,23	0	0,03	2,73	7,27	2,67	0,07	0,17	3,9	5,93
Calacuccia	10	7,27	0	0	2,73	7,3	2,67	0,03	0,03	4,03	5,93
Calvi	10	5,8	0	0,17	4,03	5,77	4,03	0,2	1,13	3,4	5,47
Ile Rousse	10	7,83	0	0	2,17	7,9	2,1	0	1	4,3	4,7
Pietralba	10	7,57	0	0	2,43	7,6	2,4	0	2,03	0,3	7,67
Bastia	10	7,23	0	0,1	1,03	6,87	1,13	0	1,33	2,03	6,63
San Pietro Tenda	8	6,87	0	0,1	1,03	6,87	1,13	0	0,57	1,83	5,6
Oletta	8	7,43	0	0	0,57	7,43	0,57	0	1,07	0,57	6,37
Cagnano	10	5,2	0	0,03	4,77	5,27	4,63	0,03	1	2,37	6,63
Cap Corse	10	3,27	0	0	6,73	3,3	6,33	0,37	0,9	3,43	3,57

Tabella A.4: Delaunay-oriented,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	11	10,3	0	0,2	0,5	10,4	0,6	0	0,17	6,5	4,33
Alistro	11	11	0	0	0	11	0	0	1,3	5,8	3,9
Corte	9	9	0	0	0	9	0	0	2,47	1,87	4,67
Calacuccia	9	9	0	0	0	9	0	0	2,73	1,5	4,77
Calvi	10	10	0	0	0	10	0	0	2,23	3,93	3,83
Ile Rousse	8	8	0	0	0	8	0	0	1,43	2,57	4
Pietralba	6	6	0	0	0	6	0	0	1,4	0,37	4,23
Bastia	9	9	0	0	0	9	0	0	1,97	3,07	3,97
San Pietro Tenda	7	7	0	0	0	7	0	0	2	0,67	4,33
Oletta	7	7	0	0	0	7	0	0	2,13	0,97	3,9
Cagnano	9	8,3	0	0,13	0,57	8,47	0,53	0	0	1,83	7,17
Cap Corse	10	9,2	0	0,03	0,77	9,2	0,8	0	0,03	5,07	4,9

Tabella A.5: Ibrido,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	6	1,63	0	0,07	4,3	1,63	4,37	0	0	2,93	3,07
Alistro	10	7	0	0,07	3,93	7	4	0	0,13	3,67	7,2
Corte	11	7,43	0	0	2,57	7,43	2,57	0	0,07	3,6	6,33
Calacuccia	11	7,5	0	0	2,5	7,43	2,57	0	0,1	3,8	6,1
Calvi	9	7,4	0	0,07	2,53	7,5	2,5	0	1,3	3,1	5,6
Ile Rousse	10	9,03	0	0,07	0,9	9,03	0,97	0	1,23	3,97	4,8
Pietralba	11	8,9	0	0	0,43	8,93	1,07	0	1,53	2,4	6,07
Bastia	11	8,27	0	0,07	1,67	8,37	1,63	0	1,17	2,27	6,57
San Pietro Tenda	10	7,73	0	0	0,27	7,77	0,23	0	0,77	0,43	6,8
Oletta	11	8	0	0	1	8	1	0	1,07	1,2	6,73
Cagnano	10	6,73	0	0,03	3,23	6,77	3,23	0	1	0,97	8,03
Cap Corse	8	6,47	0	0,07	3,47	6,4	3,6	0	1	1,4	7,6

Tabella A.6: Ibrido,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 16 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	11	8,07	0	0,13	2,8	8,4	2,6	0	0,07	4,03	6,9
Alistro	11	10,07	0	0,1	0,83	10,13	0,87	0	0,13	3,4	7,47
Corte	9	8,6	0	0,07	0,33	8,67	0,33	0	0,73	1,73	6,53
Calacuccia	9	8,2	0	0,07	0,73	8,23	0,77	0	1,03	4,03	3,93
Calvi	11	8,9	0	0,33	1,77	9,13	1,83	0,03	0,37	4,23	6,4
Ile Rousse	9	7,97	0	0,17	0,87	8,13	0,87	0	0,67	1,73	6,6
Pietralba	6	5,97	0	0	0,03	5,97	0,03	0	0,6	0,4	5
Bastia	11	10,5	0	0	0,5	10,47	0,53	0	0,63	3,17	7,2
San Pietro Tenda	8	7,87	0	0	0,13	7,83	0,17	0	0,8	0,87	6,33
Oletta	9	8,93	0	0	0,07	8,93	0,07	0	1,33	0,9	6,77
Cagnano	11	9,27	0	0,13	0,13	8,93	1,67	0	0,83	3,27	6,9
Cap Corse	11	7,3	0	0,43	3,27	7,5	3,47	0,03	0,23	3,03	7,73

Tabella A.7: Voronoi-oriented,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	6	1,97	0	0,07	3,97	1,97	3,97	0,07	0	1,77	4,23
Alistro	10	5,77	0	0	4,23	5,77	3,83	0,4	0	2,33	7,67
Corte	11	7,93	0	0	3,07	7,9	2,77	0,33	0	2,63	8,37
Calacuccia	11	5,8	0	0	5,2	5,83	4,43	0,73	0,17	2,2	8,63
Calvi	9	2,83	0	0,03	6,13	2,87	5,53	0,6	0,17	2,5	6,33
Ile Rousse	10	5,87	0	0	4,13	5,9	3,57	0,53	0,4	2,53	7,07
Pietralba	11	8,4	0	0	2,6	8,4	2,57	0,03	0,87	1,47	8,67
Bastia	11	6,67	0	0,03	4,3	6,67	3,83	0,5	0	2,53	8,47
San Pietro Tenda	10	8,07	0	0,07	1,87	8,17	1,73	0,1	0,53	1,87	7,6
Oletta	11	8,03	0	0	2,97	8,07	2,87	0,07	1,6	1	8,4
Cagnano	10	4,53	0	0,03	5,43	4,53	4,7	0,77	0,17	2,3	7,53
Cap Corse	8	2,03	0	0	5,97	2,1	4,43	1,47	0	1,5	6,5

Tabella A.8: Voronoi-oriented,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	11	8,23	0	0,4	2,37	8,53	2,47	0	0,03	7,07	3,9
Alistro	11	10,43	0	0,13	0,43	10,57	0,43	0	0,43	6,7	3,87
Corte	9	8,7	0	0	0,3	8,77	0,23	0	0,83	4,03	4,13
Calacuccia	9	8,77	0	0,03	0,2	8,77	0,23	0	0,37	2,2	6,4
Calvi	10	8,8	0	0,17	1,03	8,9	1,1	0	0,8	4,67	4,53
Ile Rousse	8	7,37	0	0,03	0,27	7,43	0,57	0	0,77	3,1	4,13
Pietralba	6	6	0	0	0	6	0	0	1,8	1,2	3
Bastia	9	8,83	0	0,03	0,13	8,9	0,1	0	1,87	2,03	5,1
San Pietro Tenda	7	6,97	0	0,03	0	6,97	0,03	0	1,97	0,77	4,27
Oletta	7	6,93	0	0,03	0,03	6,97	0,03	0	1,2	1,43	4,37
Cagnano	9	7,77	0	0,07	1,17	7,83	1,17	0	0,8	3,73	4,47
Cap Corse	10	7,83	0	0,23	1,93	7,93	2,07	0	1,23	4,1	4,67

Tabella A.9: Delaunay-oriented,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	5	1,97	0	0,07	3,83	1,97	3,97	0,07	0,03	2,93	2,03
Alistro	11	4,9	0	0,03	6,07	4,93	5,5	0,57	0,13	4,7	6,17
Corte	10	6,5	0	0,03	3,47	6,5	3,07	0,43	0	3,97	6,03
Calacuccia	10	6,67	0	0,07	3,27	6,67	3,1	0,23	0,1	3,9	6
Calvi	10	4,53	0	0,03	5,43	4,53	4,8	0,67	0,43	3,67	5,9
Ile Rousse	10	7,07	0	0,1	2,83	7,1	2,77	0,13	0,53	3,43	6,03
Pietralba	10	7	0	0	3	7	3	0	1,77	0,57	7,67
Bastia	10	6,63	0	0,03	3,33	6,63	3,2	0,17	1,2	1,97	6,83
San Pietro Tenda	8	6,33	0	0,1	1,57	6,33	1,67	0	0,43	1,43	6,13
Oletta	8	6,63	0	0,03	1,33	6,63	1,37	0	0,9	0,47	6,63
Cagnano	10	4,93	0	0,07	5	4,93	4,63	0,43	0,93	1,83	7,23
Cap Corse	10	2,77	0	0,1	7,13	2,8	5,87	1,33	0,9	3,23	5,87

Tabella A.10: Delaunay-oriented,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	11	9,2	0	0,1	1,7	9,23	1,77	0	0,2	6,27	4,53
Alistro	11	10,93	0	0,07	0	11	0	0	1,8	4,2	5
Corte	9	8,97	0	0,03	0	9	0	0	1,97	1,97	5,1
Calacuccia	9	9	0	0	0	9	0	0	1,6	2,3	5,1
Calvi	10	9,83	0	0,07	0,1	9,87	0,13	0	0,07	6	3,93
Ile Rousse	8	7,97	0	0,03	0	8	0	0	0,8	3,2	4
Pietralba	6	6	0	0	0	6	0	0	0,47	0,5	4,37
Bastia	9	9	0	0	0	9	0	0	1,67	2,33	5
San Pietro Tenda	7	7	0	0	0	7	0	0	1,97	0,43	4,6
Oletta	7	7	0	0	0	7	0	0	2,2	0,5	4,3
Cagnano	9	8,13	0	0,03	0,83	8,17	0,83	0	0	2,67	6,33
Cap Corse	10	8,97	0	0,07	0,97	9,03	0,97	0	0,1	5,43	4,47

Tabella A.11: Ibrido,  $B = 5000 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

Sorgente	N. GRAY	OSP				DSP			GSP		
		S	F	D	C	S	D	F	S	D	F
Solenzara	6	1,57	0	0,1	4,33	1,57	4,43	0	0	3,03	2,97
Alistro	11	6,53	0	0	4,47	6,57	4,27	0,17	0,13	4,37	6,5
Corte	10	7,3	0	0,03	2,67	7,3	2,63	0,07	0	3,47	6,53
Calacuccia	10	7,17	0	0,13	2,7	7,17	2,83	0	0	3,57	6,43
Calvi	10	6,03	0	0	3,97	6,03	3,83	0,13	0	4	6
Ile Rousse	10	7,9	0	0	2,1	7,93	2,07	0	0,43	3,1	5,6
Pietralba	10	8,03	0	0,07	1,9	8,07	1,93	0	1,7	1,03	7,27
Bastia	10	7,43	0	0,07	2,5	7,47	2,53	0	1,2	1,83	6,97
San Pietro Tenda	8	7,07	0	0,03	0,9	7,07	0,93	0	0,8	0,33	6,87
Oletta	9	7,83	0	0	1,17	7,83	1,17	0	1,07	1,17	6,77
Cagnano	10	5,3	0	0,03	4,67	5,37	4,63	0	1	0,67	8,33
Cap Corse	10	3,7	0	0	6,3	3,7	5,93	0,37	1	2,57	6,43

Tabella A.12: Ibrido,  $B = 2500 \text{ kJ}$ ,  $m_p = 2 \text{ kg}$  e 8 classi di vento.

# Elenco delle figure

2.1	Esempio dei due principali modelli di drone: (a) fixed-wing drone, (b) quadcopter.	7
2.2	Esempio dei due principali modelli di drone ibridi: (a) convertiplano, (b) tail-sitter	8
2.3	Schema riassuntivo dei diversi parametri di valutazione di un drone.	10
2.4	Classificazione delle diverse operazioni realizzabili dai droni.	13
2.5	Schema di una rete WSN, in cui le frecce dritte rappresentano gli spostamenti del drone e le frecce ondulate raffigurano le comunicazione.	17
2.6	Illustrazione delle strategie di allocazione delle comunicazione per dispositivi mobili con l'ausilio dei droni.	18
3.1	Direzione del vento relativo $\omega_d(e)$ . Modello illustrativo in cui l'arco $e$ è in blu e la direzione del vento globale è in rosso.	24
3.2	Schemi riassuntivi della semplificazione in classi della direzione del vento relativo $\omega_d(e)$ .	25
3.3	Grafico dei consumi energetici unitari di un drone con velocità media variabile $v_d$ .	28
3.4	Rappresentazione di un possibile prospetto di smart-city.	30
3.5	Tassellazione di Voronoi di un insieme di punti $P$ .	31
3.6	Triangolazione di Delauney di un insieme di punti $P$ .	34
3.7	Collegando i centri dei circumcerchi si ottiene il diagramma di Voronoi (in rosso).	35
3.8	Illustrazione di un grafo Voronoi-Oriented.	37
3.9	Esempio di calcolo dei consumi di un grafo generico.	38
3.10	Esempio di associazione dei venti su un lato di una regione di Voronoi.	39

3.11	Esempio di un'istanza di grafo in cui i lati di Delaunay attraversano più di due regioni di Voronoi. . . . .	40
3.12	Esempio di un'istanza di soluzione in cui i lati di Delaunay attraversano più di due regioni di Voronoi. . . . .	41
3.13	Esempio di associazione dei venti su un lato della triangolazione di Delaunay. . . . .	42
4.1	Grafo time-dependent che non ammette una soluzione fattibile da $S$ a $D$ . Qui, i costi degli archi per la sotto-sequenza dei grafi "pari" $G_0, G_2, G_4$ sono raffigurati in (a), mentre i costi per la sotto-sequenza dei grafi "dispari" $G_1, G_3, G_5$ sono raffigurati in (b). La sorgente del deposito $s$ è raffigurata in blu, mentre la destinazione $d$ è raffigurata in rosso. . . . .	50
5.1	Mappa che descrive la disposizione delle centraline meteo sul suolo della regione Corsica. . . . .	54
5.2	Esempio di alcune rilevazioni fatte dalla centralina di Alistro. . . . .	55
5.3	Differenti grafi. . . . .	57
5.4	Grafo completo costruito considerando come vertici le stazioni meteo della Corsica. . . . .	58
5.5	Grafi pesati con le distanze espresse in chilometri . . . . .	59
5.6	Esempi di missioni con $B = 10$ . La freccia indica dove si trova il drone. gli archi tratteggiati e continui rappresentano, rispettivamente, gli archi pianificati (non percorsi) e quelli percorsi. Il deposito è indicato con il blu e la destinazione con il rosso. In dettaglio: CANCELED (solo OSP) in (a); DELIVERED in (b); FAIL in (c); SUCCESS in (d). . . . .	61
5.7	Risultati Pre-Processing rispetto al numero complessivo di vertici . .	65
5.8	Risultati delle simulazioni che descrivono in percentuale gli stati ottenuti fissato $B = 5000$ kJ. . . . .	68
5.9	Risultati delle simulazioni che descrivono in percentuale gli stati ottenuti fissato $B = 2500$ kJ. . . . .	70

# Elenco delle tabelle

3.1	Tabella dei consumi unitari espressi in kJ, con $m_p = 0 \text{ kg}$ e $v_d = 0 \text{ m/s}$ .	28
3.2	Tabella dei consumi unitari espressi in kJ, con $m_p = 2 \text{ kg}$ e $v_d = 0 \text{ m/s}$ .	29
3.3	Tabella dei consumi unitari espressi in kJ, con $m_p = 7 \text{ kg}$ e $v_d = 0 \text{ m/s}$ .	29
5.1	Tabella dei cammini minimi calcolati con sorgente in Solenzara.	60
5.2	Tabella dei risultati del Pre-Processing con sorgente in Solenzara e batteria con 5000 kJ per i differenti modelli di grafo.	62
5.3	Confronto tra OSP, DSP, e GSP su dati quasi-reali. C, S, D, e F rappresentano CANCELED, SUCCESS, DELIVERED, e FAIL, rispettivamente. Dati sono espressi in percentuale.	72
A.1	Voronoi-oriented, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	75
A.2	Voronoi-oriented, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	76
A.3	Delaunay-oriented, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	76
A.4	Delaunay-oriented, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	77
A.5	Ibrido, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	77
A.6	Ibrido, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 16 classi di vento.	78
A.7	Voronoi-oriented, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	78
A.8	Voronoi-oriented, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	79
A.9	Delaunay-oriented, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	79
A.10	Delaunay-oriented, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	80
A.11	Ibrido, $B = 5000 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	80
A.12	Ibrido, $B = 2500 \text{ kJ}$ , $m_p = 2 \text{ kg}$ e 8 classi di vento.	81

# Bibliografia

- [1] Maziar Arjomandi, Shane Agostino, Matthew Mammone, Matthieu Nelson, and Tong Zhou. Classification of unmanned aerial vehicles. *Report for Mechanical Engineering class, University of Adelaide, Adelaide, Australia*, 2006.
- [2] Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, 2004.
- [3] P. Bhattacharya and M. L. Gavrilova. Voronoi diagram in optimal path planning. In *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, pages 38–47, 2007.
- [4] Taula M Cabreira, Lisane B Brisolara, and Paulo R Ferreira Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019.
- [5] Matthew Clothier and Mike Bailey. Creating destructible objects using a 3d Voronoi subdivision tree. In *2015 15th International Conference on Computational Science and Its Applications*, pages 42–46. IEEE, 2015.
- [6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. ISBN 978-0-262-03384-8. URL <http://mitpress.mit.edu/books/introduction-algorithms>.
- [7] Mark de Berg et al. *Computational geometry: algorithms and applications, 2nd Edition*. Springer, 2000. ISBN 3540656200. URL <https://www.worldcat.org/oclc/43076913>.
- [8] DHL. Dhl parcelcopter launches initial operations for research purposes, 2020. [https://www.dhl.com/en/press/releases/releases\\_2014/group/dhl\\_](https://www.dhl.com/en/press/releases/releases_2014/group/dhl_)

`parcelcopter_launches_initial_operations_for_research_purposes.html_japanlithiumbatteryxml.html`.

- [9] Martin Erwig. The graph Voronoi diagram with applications. *Networks: An International Journal*, 36(3):156–163, 2000.
- [10] Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.
- [11] Steven Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1-4):153, 1987.
- [12] Suraj G Gupta, Dr Ghonge, Pradip M Jawandhiya, et al. Review of unmanned aircraft system (uas). *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, 2, 2013.
- [13] Wayne Johnson. *Helicopter theory*. Courier Corporation, 2012.
- [14] J Mark Keil and Carl A Gutwin. The Delaunay triangulation closely approximates the complete euclidean graph. In *Workshop on Algorithms and Data Structures*, pages 47–56. Springer, 1989.
- [15] Patchara Kitjacharoenchai, Mario Ventresca, Mohammad Moshref-Javadi, Seokcheon Lee, Jose MA Tanchoco, and Patrick A Brunese. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, 129:14–30, 2019.
- [16] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3):345–358, 1992.
- [17] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.
- [18] Michał Mazur, A Wisniewski, and Jeffery McMillan. Clarity from above: Pwc global report on the commercial applications of drone technology. *Warsaw: Drone Powered Solutions, PriceWATER house Coopers*, 2016.

- [19] Max Mühlhäuser, Christian Meurisch, Michael Stein, Jörg Daubert, Julius Von Willich, Jan Riemann, and Lin Wang. Street lamps as a platform. *Communications of the ACM*, 63(6):75–83, 2020.
- [20] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018.
- [21] Adnan S Saeed, Ahmad Bani Younes, Shafiqul Islam, Jorge Dias, Lakmal Seneviratne, and Guowei Cai. A review on the platform design, dynamic modeling and control of hybrid uavs. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 806–815. IEEE, 2015.
- [22] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 151–162. IEEE, 1975.
- [23] Francesco Betti Sorbelli, Federico Coro, Sajal K. Das, and Cristina M. Pinotti. Energy constrained delivery of goods with drones under varying wind conditions. Private Communication, 2020.
- [24] Joshuah K Stolaroff, Constantine Samaras, Emma R O’Neill, Alia Lubers, Alexandra S Mitchell, and Daniel Ceperley. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature communications*, 9(1):1–13, 2018.
- [25] Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.
- [26] KE Trummel and JR Weisinger. The complexity of the optimal searcher path problem. *Operations Research*, 34(2):324–327, 1986.
- [27] Yishu Wang, Ye Yuan, Yuliang Ma, and Guoren Wang. Time-dependent graphs: Definitions, applications, and algorithms. *Data Science and Engineering*, 4(4):352–366, 2019.

# Ringraziamenti

Desidero vivamente ringraziare il mio relatore, la Prof.ssa Maria Cristina Pinotti per la disponibilità e la professionalità dimostratami in questi mesi di lavoro. È stata un esempio di come sia possibile amare e far amare lo studio e la conoscenza di una materia non sempre facile. È un bagaglio, sia culturale che di impostazione mentale, che porterò ovunque con me.

Ringrazio la mia famiglia, il primo punto di riferimento, che mi ha sostenuto sia economicamente che emotivamente e che mi ha permesso di percorrere e concludere questo cammino.

Ringrazio Giorgia, per il tempo dedicato ad ascoltare i miei interminabili ragionamenti e per la pazienza dimostrata in questo intenso periodo.

Infine vorrei ringraziare una persona unica e speciale, che ora non è più tra noi, il Prof. Andrea Caporali che ha saputo trasmettermi l'amore per l'informatica, insegnandomi a non porre limiti alla mia fantasia e alle mie idee.