Università degli Studi di Perugia
Tesi Magistrale in Informatica

L'instradamento degli octocopter alla luce delle condizioni del vento

Autore:
Lorenzo Palazzetti

Relatore:
Prof.ssa Maria Cristina Pinotti

Anno Accademico 2019/2020

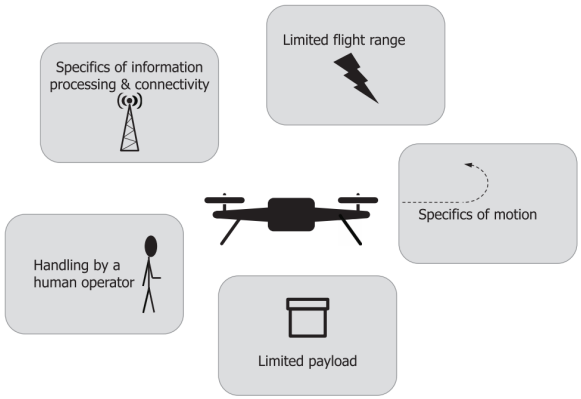
Outline

Outline

Classificazione dei droni

I droni possono essere classificati sulla base dei seguenti parametri:

- Proprietà di movimento
- Carico
- Durata del volo
- Connettività



Principali tipi di drone:

(a) Rotorcraft.

(b) Fixed-wing.

(c) Tail-Sitter.

(d) Convertplane.

Applicazioni in ambito civile

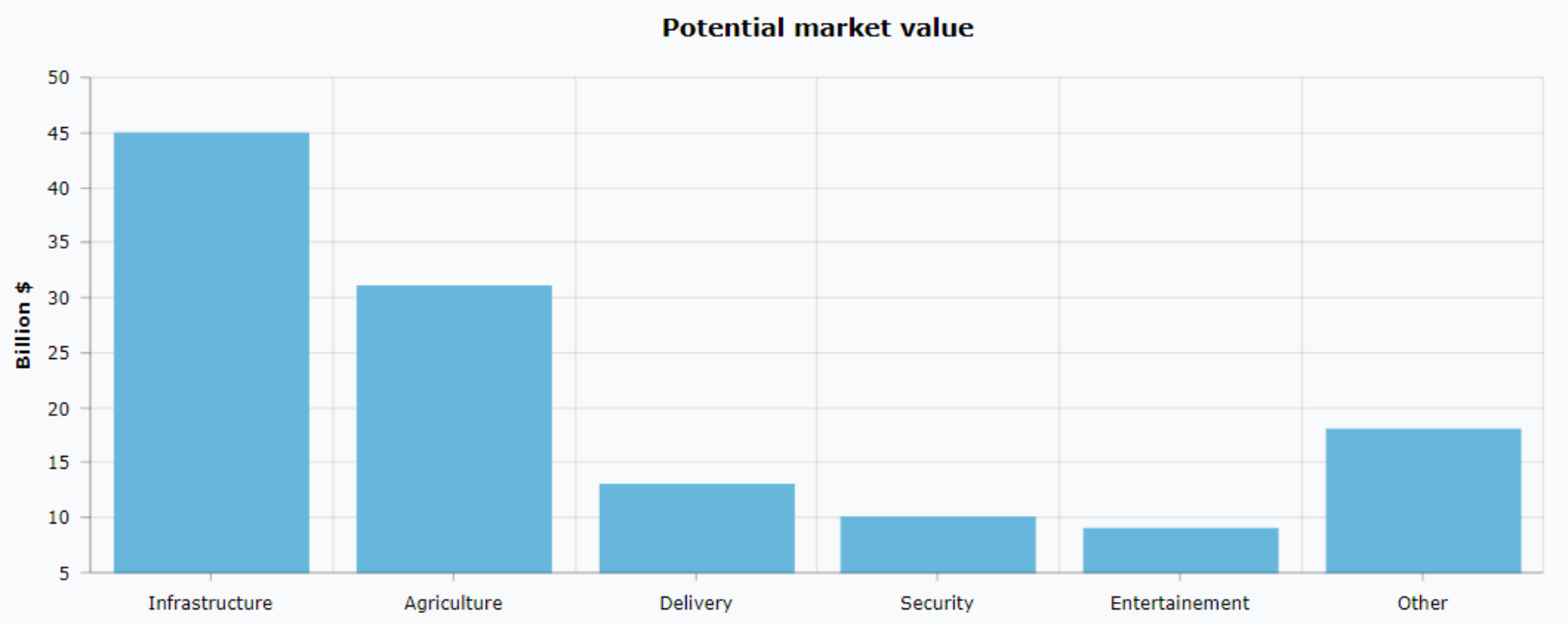
(e) Agricoltura.

(f) Consegne.

(g) Sorveglianza.

(h) Film making.

Applicazioni in ambito civile



Outline

Problemi di consegna

- I problemi di consegna con mezzi tradizionali sono risolti riducendoli a problemi combinatorici classici come il *travelling salesman problem* e il *vehicle routing problem*
- Le restrizioni di carico imposte dall'octocopter rendono inappropriata una tale soluzione. Per i droni, si definisce invece uno schema di consegna di tipo **end-to-end**.
- Sia $G = (V, E)$ la mappa di routing dove
  - l'insieme dei vertici $V = \{v_0, v_1, ..., v_n\}$ è la posizione degli end-customer
  - l'insieme degli archi $E = \{e_0, e_1, ..., e_m\}$ definisce le vie percorribili
  - la funzione costo $W : E \rightarrow \mathbb{R}^+$ definisce i consumi energetici in relazione alle **condizioni di vento locali**
- Dati
  - la sorgente $s \in V$ e la destinazione $d \in V$
  - il budget B
- Il problema di consegna End-to-End (EEP) richiede di trovare il ciclo C che inizia e termina in s passando per d di costo **non superiore** a B, considerando le variazioni di vento locale.

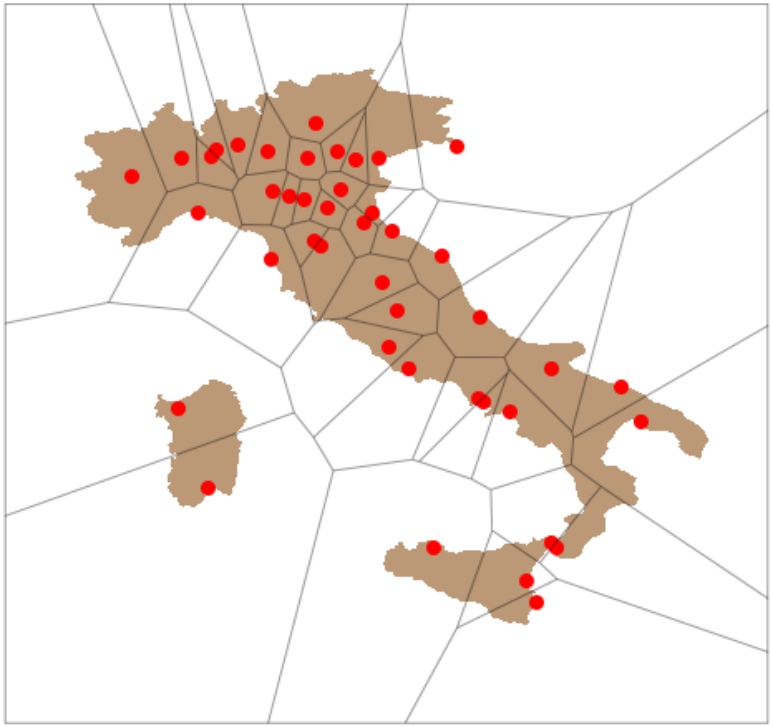
Definizione modello per i grafi

- Dati
  - Uno spazio metrico X caratterizzato da una funzione distanza d
  - Un insieme di siti $\mathcal{S} = \{s_0, s_1, ..., s_n\}$
  - Un insieme discreto di venti globali $\Omega = \{\omega_0, \omega_1, ..., \omega_n\}$
- Si ha che ogni punto $s_i \in \mathcal{S}$ è caratterizzato da un vento globale $\omega_i \in \Omega$ delimitato dai lati della **cella di Voronoi** R_i associata al sito s_i

Definizione: Diagramma di Voronoi

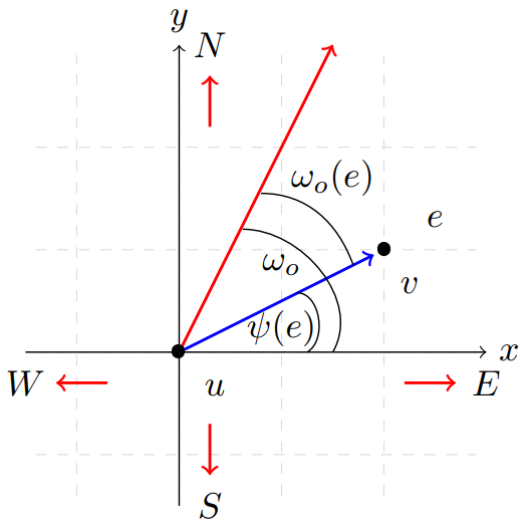
La cella di Voronoi, o regione di Voronoi, R_k, associata al sito s_k è l'insieme di tutti i punti in X le cui distanze da s_k non sono più grandi della loro distanza rispetto ad altri siti s_j, dove $j \neq k$.

Esempio di tassellazione di Voronoi su un insieme di siti



Definizione di vento relativo

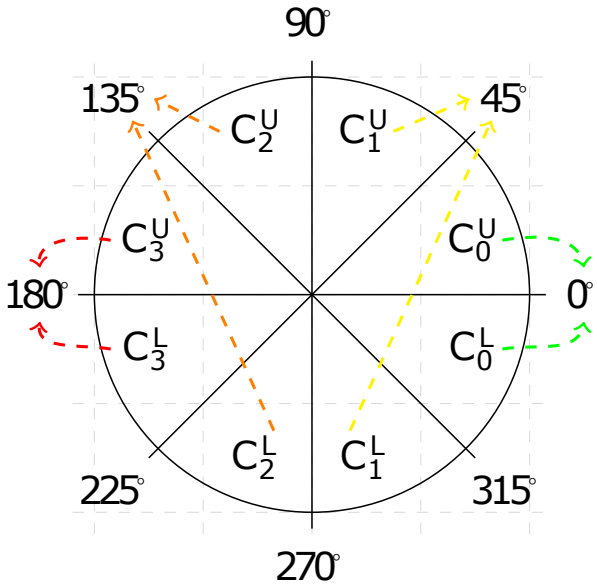
- La funzione costo d_t dalla nozione di *vento globale* $\omega = (\omega_s, \omega_d)$
  - ω_s è la velocità di ω
  - ω_d è la direzione di ω



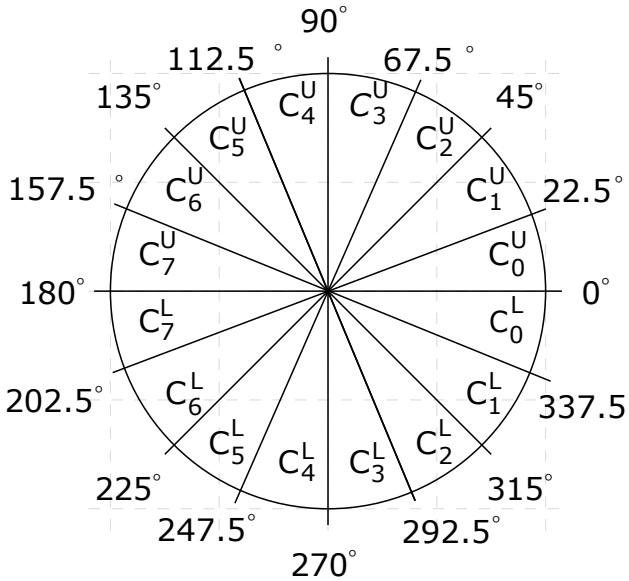
- Dati:
  - Un arco $e = (u, v)$
  - La condizione di vento globale $\omega \in \Omega$
  - L'angolo relativo dell'arco (u, v), $\psi(e) = arctg(\frac{y_v}{x_v})$
- La *direzione del vento relativo* può essere definita come $\omega_d(e) = \omega_d - \psi(e)$

Definizione di vento relativo

- La *discretizzazione* dei venti relativi si ottiene semplificando le occorrenze della direzione relativa del vento raggruppando i valori di $\omega_d(e)$ in 8 e 16 settori



(i) Raggruppamento in 8 settori.



(j) Raggruppamento in 16 settori.

Definizione dei consumi energetici

- Dati
  - m_p, la massa del carico trasportato
  - s_d, la velocità del drone
  - $\omega_d(e)$, la direzione del vento relativo
  - ω_s, la velocità del vento
- Il costo energetico di attraversamento dell'arco e all'istante t può essere definito come

$$\eta(e) = \mu(e)\lambda(e). \tag{1}$$

- Dove
  - $\mu(e)$ è il consumo energetico unitario
  - $\lambda(e)$ è la lunghezza dell'arco e

Esempio di consumi unitari

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

(k) Tabella consumi unitari(kJ) $m_p = 0$ kg e $s_d = 10$ m/s.

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

(l) Tabella consumi unitari(kJ) $m_p = 2$ kg e $s_d = 10$ m/s.

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

(m) Tabella consumi unitari(kJ) $m_p = 7$ kg e $s_d = 10$ m/s.

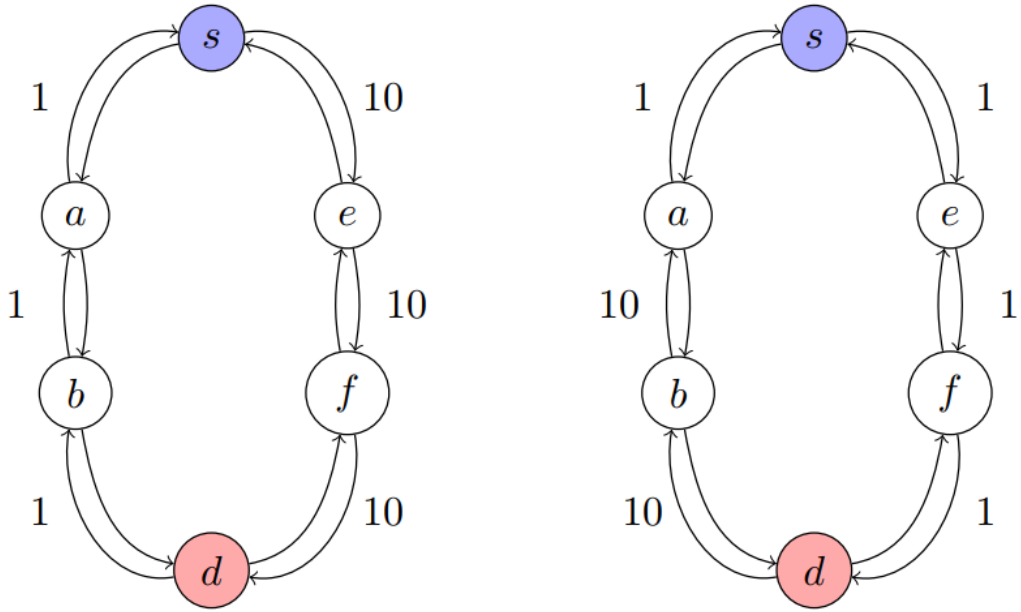
Mission Feasibility Problem

- Time-dependent cost delivery network graph $\mathcal{G} = \{\mathcal{G}_0, ..., \mathcal{G}_t\}$
  - E' definito da una serie di grafi o *snapshot* statici $\mathcal{G}_t$
  - Per ogni istanza $\mathcal{G}_i, i = 0, 1, ..., n$ la funzione peso varia in riferimento alla nozione di *vento globale*
- Time-dependent $\mathcal{G}_t = (V, E; \eta_t)$
  - L'insieme dei vertici $V = \{v_0, v_1, ...v_n\}$
  - L'insieme degli archi $E = \{e_0, e_1, ...e_m\}$
  - $\eta_t : E \to \mathbb{R}^+$ è la funzione costo energetico all'instante di tempo $t \in \mathbb{N}$

Definizione: Mission Feasibility Problem (MFP)

Il *Mission-Feasibility Problem* è un problema decisionale che richiede di stabilire se EEP ammette soluzioni, dato $\mathcal{G}$, il budget B, il payload m_p, la velocità del drone s_d, la sorgente s, la destinazione d e l'orario di partenza i

Time-dependent cost delivery network graph $\mathcal{G}$



(a) G_0. (b) G_1.

Figura: 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.

Outline

Algoritmi per MFP

- Per risolvere e studiare il MFP sono stati definiti i seguenti algoritmi:
  - Offline Shortest Path (OSP)
  - Dynamic Shortest Path (DSP)
  - Greedy Shortest Path (GSP)
- Confrontando il consumo energetico ottenuto con il budget a disposizione viene restituito uno dei seguenti stati finali:
  - SUCCESS (S)
  - DELIVERED (D)
  - FAIL (F)
  - CANCELED (C)

Algoritmo di Offline Shortest Path (OSP)

- L'algoritmo OSP risolve il EEP valutando i costi all'instante t =0 e calcolando il ciclo C nel seguente modo:
  1. Calcola il cammino minimo da s a d, $\pi_{s,d}$
  2. Calcola il cammino minimo da d a s, $\pi_{d,s}$
- I cammini minimi vengono computati utilizzando l'algoritmo di *Dijkstra*
- Durante l'esecuzione, i costi energetici sono aggiornati usando $\mathcal{G}$
- Il costo computazionale complessivo di OSP è $\mathcal{O}(|E| + |V| \log |V|)$

Algoritmo di Dynamic Shortest Path (DSP)

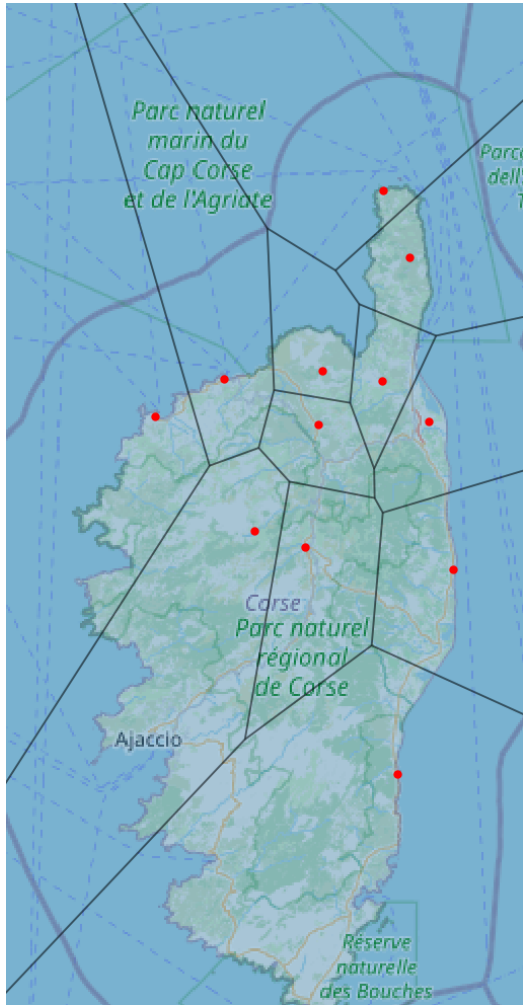
- L'algoritmo DSP risolve il EEP calcolando il ciclo C nel seguente modo:
  1. Calcola il cammino minimo da $s = u_0$ a d, $\pi_{s,d}$, valutando i costi in G_0 e aggiunge alla soluzione l'arco $e_0 = (u_0, u_1)$
  2. Per ogni nodo intermedio u_i valutando i costi in G_i ricalcola il cammino minimo $\pi_{u_i,d}$ e aggiunge l'arco $e_i = (u_i, u_{i+1})$ fintanto che non si raggiunge il nodo d
  3. Ripete il processo per trovare il cammino inverso $\pi_{d,s}$
- I cammini minimi vengono computati utilizzando l'algoritmo di *Dijkstra*
- Il costo computazionale complessivo di DSP è $\mathcal{O}(|C|(|E| + |V| \log |V|))$

Algoritmo di Greedy Shortest Path (GSP)

- L'algoritmo GSP risolve il EEP calcolando il ciclo C nel seguente modo:
  1. Calcola il cammino da s a d, $\pi_{s,d}$, attraversando ad ogni passo l'arco $e = (u_i, u_{i+1})$ di costo minimo sul $\mathcal{G}_i$
  2. Ripete lo stesso procedimento per trovare il cammino da d a s,$\pi_{d,s}$
- Il costo computazionale complessivo di GSP è $\mathcal{O}(|C|(\max_{v \in V} |Adj(v)|))$

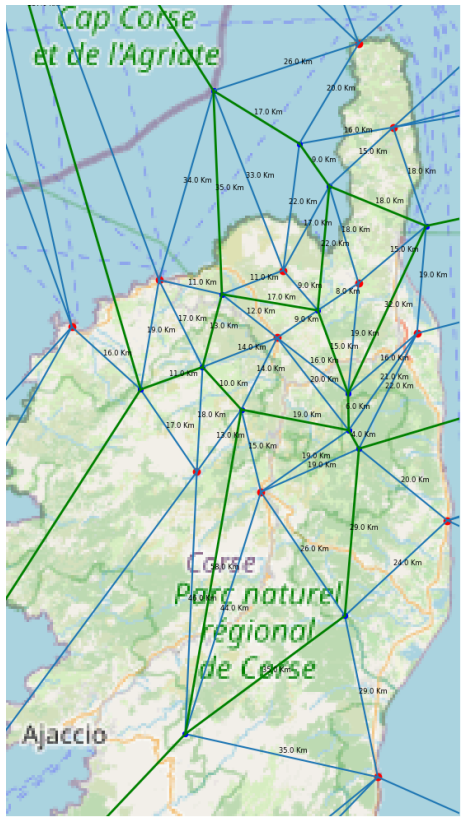
Outline

Descrizione dei dati



Grafo Voronoi-oriented

- $\mathcal{S} = \{s_0, s_1, ..., s_k\}$ l'insieme delle centraline
- $Vor(\mathcal{S})$ la tassellazione di Voronoi dei punti $\mathcal{S}$ dove
  - L'insieme degli *archi di Voronoi* è $E_{vor} = \{e_0, e_1, ..., e_j\}$ con $j \leq 3k - 6$
  - L'insieme dei *vertici di Voronoi* è $V_{vor} = \{v_0, v_1, ..., v_j\}$ con $j \leq 2k - 5$
- Si definisce inoltre $E_{cent} = \{(s, v) : s \in \mathcal{S}, v \in Reg(s)\}$
- Il grafo **Voronoi-oriented** $G_{VO} = (V, E)$
  - $V = \mathcal{S} \cup V_{vor}, \quad |V| =$34
  - $E = E_{cent} \cup E_{vor}, \quad |E| =$42
  - $d(G) =$6



Grafo Voronoi-oriented

Figura: Illustrazione consumo energetico per archi che collegano il centro ai vertici di una cella di Voronoi.

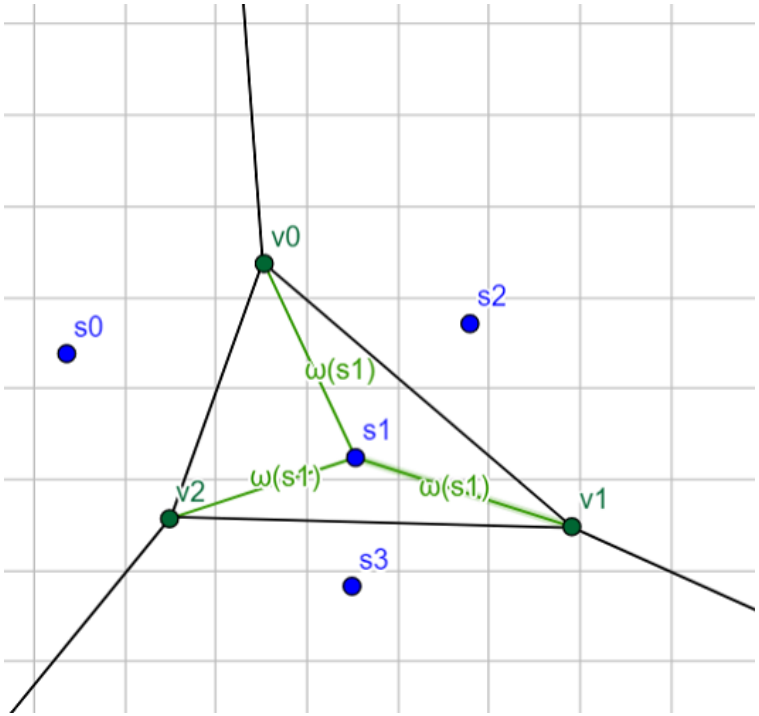
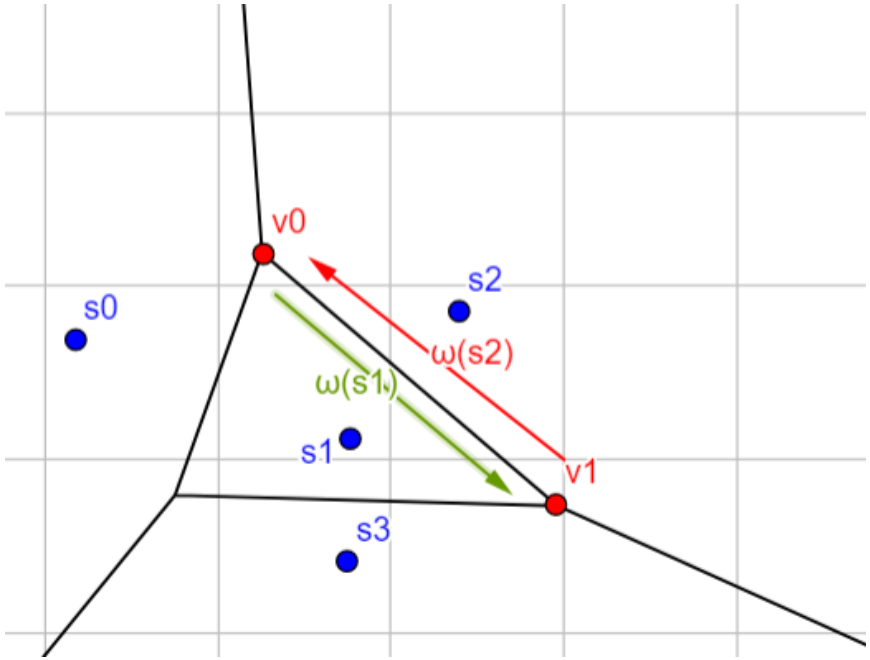
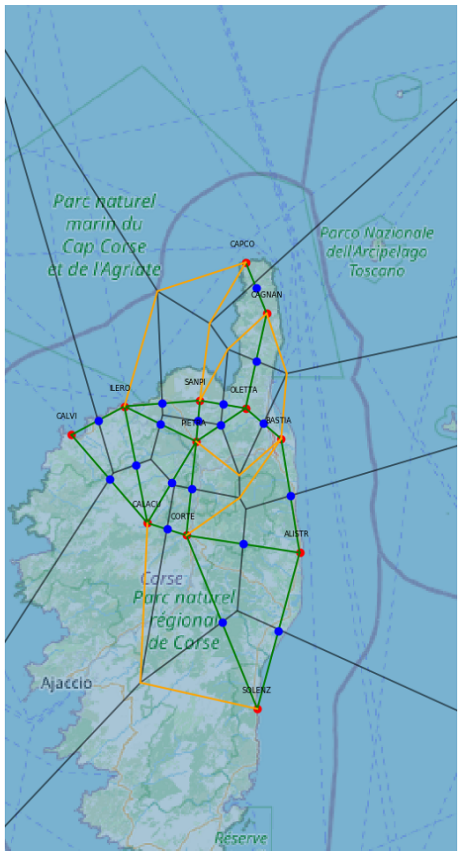


Figura: Illustrazione consumo energetico per archi di confine fra due celle di Voronoi.



Grafo Delaunay-oriented

- $\mathcal{S} = \{s_0, s_1, ..., s_k\}$ l'insieme delle centraline
- $DT(\mathcal{S})$ la triangolazione di Delaunay dei punti $\mathcal{S}$ dove
  - L'insieme degli *archi di Delaunay* è $E_{dt} = \{e_0, e_1, ..., e_i\}$ con $i \leq 3k - 3 - z$ dove $z = |Hull(\mathcal{S})|$
  - Gli *archi che giacioni in più di due celle di Voronoi* sono $E_{\bar{dt}} = \{\bar{e}_0, \bar{e}_1, ..., \bar{e}_j\}$ con $j < i$
- Per ogni $e = (s_i, s_j) \in E_{\bar{dt}}$ si aggiunge un vertice di Voronoi v_k e due archi $(s_i, v_k), (v_k, s_j) t.c.$ $\min\{d(s_i, v_k) + d(v_k, s_j)\}$
- I vertici e gli archi aggiunti definiscono V_{vor}, E_{vor}
- Il grafo **Delaunay-oriented** $G_{DO} = (V, E)$
  - $V = \mathcal{S} \cup V_{vor}, \quad |V| =$19
  - $E = E_{dt} \cup E_{vor} \setminus E_{\bar{dt}}, \quad |E| =$25
  - $d(G) =$7



Grafo Delaunay-oriented

Figura: Illustrazione consumo energetico per archi che attraversano esattamente due celle di Voronoi.

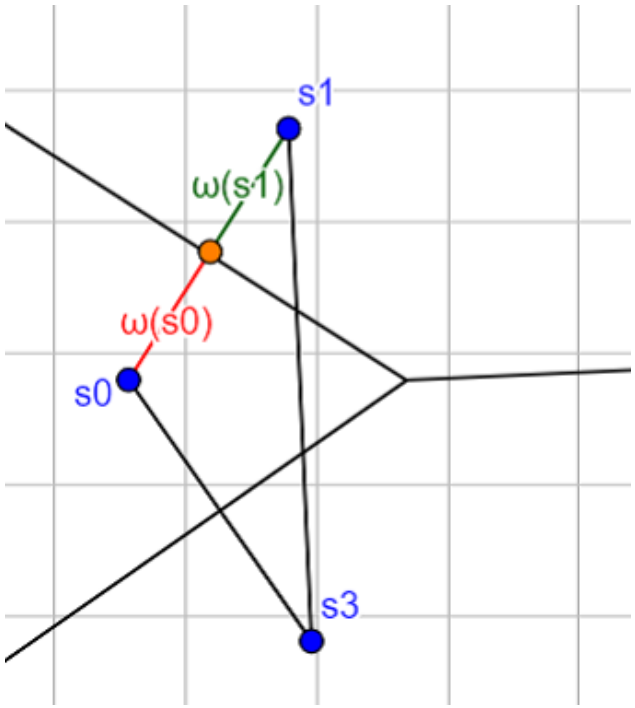
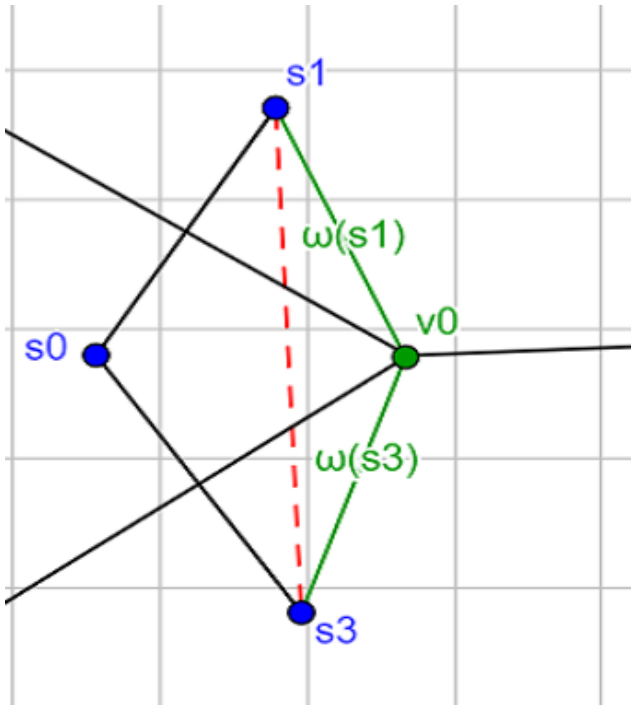
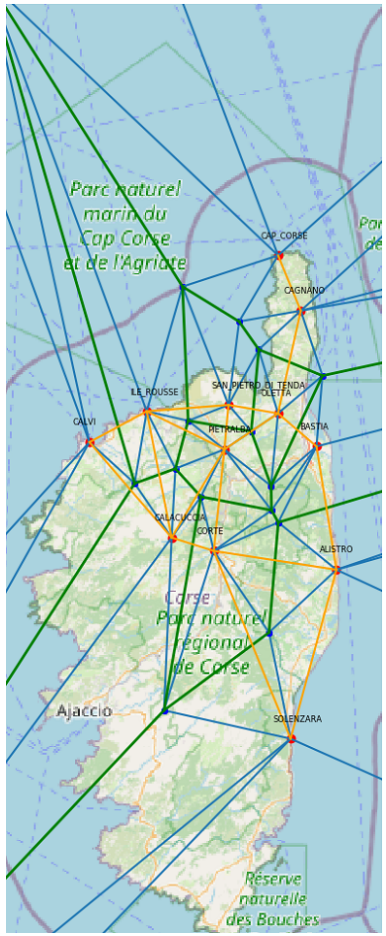


Figura: Illustrazione consumo energetico per archi che attraversano più di due celle di Voronoi.



Grafo ibrido

- Dato
  - Il grafo Voronoi-oriented $G_{VO} = (V_{VO}, E_{VO})$
  - Il grafo Delaunay-oriented $G_{DO} = (V_{DO}, E_{DO})$
- Il grafo **Ibrido** $G_{HG} = (V, E)$
  - $V = V_{VO} \cup V_{DO}, \quad |V| =$52
  - $E = E_{VO} \cup E_{DO}, \quad |E| =$60
  - $d(G) =$6
- Il grafo ibrido rappresenta la fusione della soluzione Voronoi-oriented e Delaunay-oriented



Outline

Procedura di simulazione

- Per valutare come i venti influenzino il completamento di una missione di consegna end-to-end è stata implementata la procedura NTEST
- Fissato un insieme di orari di inizio $\mathcal{H}$
  1. Scorre ogni $h \in \mathcal{H}, s \in \mathcal{S}, d \in \mathcal{S} \setminus \{s\}$
  2. Computando le procedure OSP,DSP e GSP passandogli i parametri (h, s, d)
  3. Calcolando infine per ogni algoritmo il numero medio di stati ottenuti
- Nelle simulazioni svolte $|\mathcal{H}| = 30$
- La procedura descritta è stata ripetuta sia con $|B| = 5000\,\mathrm{kJ}$ che con $|B| = 2500\,\mathrm{kJ}$

Esempio di risultato

		OSP				DSP			GSP		
Sorgente	N. GRAY	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: Voronoi-oriented, B =5000 kJ, $m_p = 2\,\mathrm{kg}$ e 16 classi di vento.

Risultati conclusivi: confronto OSP, DSP, GSP

			OSP				DSP			GSP		
	B	gray	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

[1] I dati sono espressi in percentuale.

Commento dei risultati

- I risultati migliori sono ottenuti da DSP che ha una percentuale alta di SUCCESS ed recupera i CANCELED di OSP come DELIVERED
- L'algoritmo GSP basato sulla scelta locale ottima porta risultati fortemente negativi: alta percentuale di FAIL e bassa percentuale di SUCCESS. Inoltre complessità in tempo molto alta perchè $|C| \to |E|$.
- Il numero di classi di vento influenza soprattutto DSP che ottiene una percentuale maggiore di SUCCESS
- Dimezzando il budget energetico si dimezzano le prestazioni e si marcano maggiormente le peculiarità osservate dei tre algoritmi

Commento dei risultati rispetto ai tipi di grafo utilizzati

- I risultati sono leggermente peggiori per VG che ha gli archi più lunghi
- HG che possiede la maggior connettività e determina i migliori risultati

Grazie per l'attenzione

Sviluppi futuri

1. Si potrebbero studiare soluzioni *pre-calcolate* che considerino i cambiamenti di vento più frequenti, evitando il calcolo online eseguito dal DSP
2. Si potrebbe introdurre la possibilità che durante la missione il drone si *fermi ed attenda le condizioni migliori* di vento per raggiungere la destinazione ottimizzando il consumo energetico
3. Studiare soluzioni che considerano *punti di ricarica* nel percorso, ottimizzando i tempi di consegna
4. Definire grafi con *rotte dinamiche* che sfruttano il vento favorevole al momento