



UNIVERSITÀ DI FIRENZE
DIPARTIMENTO DI INFORMATICA

Corso di Laurea Triennale in Ingegneria Informatica

*Controllo di copertura dinamico per il
monitoraggio cooperativo in tempo reale*

Relatore

Giorgio Battistelli

Correlatore

Nicola Forti

Candidato

Ivan Necerini

Anno accademico 2023 / 2024

Indice

1	Introduzione	2
1.1	Descrizione del problema	3
1.2	Formulazione matematica del problema	4
1.3	Soluzioni proposte	5
2	Algoritmi	7
2.1	Algoritmo di coverage V1: algoritmo di ascesa del gradiente classico .	7
2.2	Algoritmo V2: aggiunta di un moto Browniano	7
2.3	Algoritmo V3: aggiunta di un potenziale repulsivo	7
3	Implementazione degli algoritmi	8
3.1	Descrizione del codice (e tabelle con pseudocodice)	8
3.2	Librerie Python utilizzate	8
4	Simulazione e discussione dei risultati	9
5	Conclusioni	10

1 Introduzione

dire RL e degli altri algoritmi che esistono e utilita generali ecc...

1.1 Descrizione del problema

In questo lavoro, un piccolo gruppo di agenti mobili (*UAVs, droni*) con capacità di rilevamento limitate, posti ad un'altezza fissa, ha l'obiettivo di monitorare dinamicamente un'area popolata da un gran numero di punti di interesse (*Targets mobili*). Lo scenario di riferimento potrebbe essere, ad esempio, un'area portuale dove i targets sono rappresentati da imbarcazioni che si muovono continuamente nel tempo.

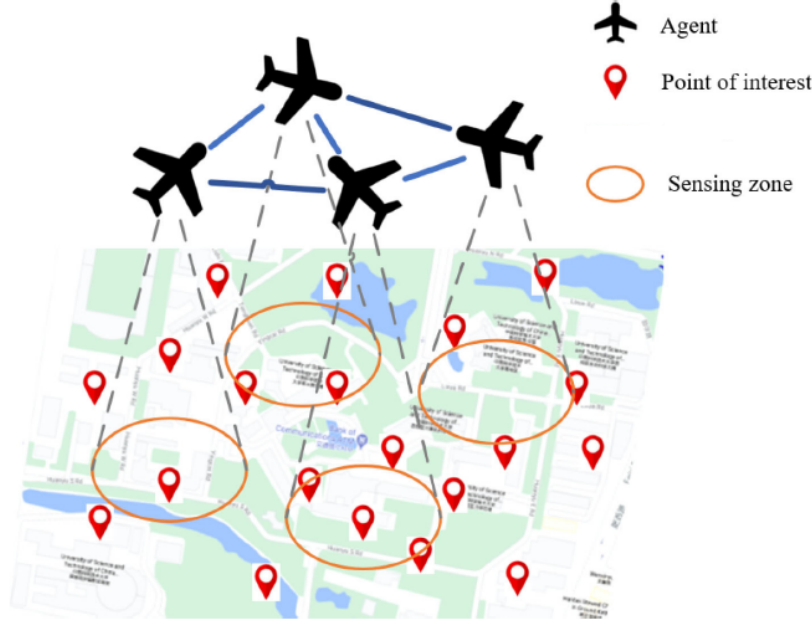


Figura 1: Possibile scenario di un controllo della copertura dinamica (2).

Il monitoraggio dinamico di un'area di interesse richiede che, istante per istante, i droni forniscano una visione accurata (o almeno sufficiente) di tutti i target mobili presenti. L'obiettivo generale può variare, come ad esempio l'evitare collisioni tra le imbarcazioni.

Per raggiungere questo obiettivo, è fondamentale garantire una *copertura sufficiente* dell'area. Nel contesto di questo esperimento, le traiettorie dei target saranno predefinite e verranno assegnate posizioni iniziali ai droni. **Il compito** consiste nell'utilizzare un algoritmo di copertura per determinare le traiettorie dei droni in modo tale che, al termine delle traiettorie dei target, venga raggiunto un grado sufficiente di copertura dell'area.

I droni dovranno quindi "seguire" i target, assicurando una buona visione dell'area di interesse per tutta la durata dell'esperimento. La definizione dettagliata del *grado di copertura sufficiente* verrà fornita nella sezione 1.3 della tesi.

1.2 Formulazione matematica del problema

Consideriamo M punti di interesse (**Targets dinamici**) distribuiti in uno spazio di lavoro bi-dimensionale W . Le posizioni dei targets sono denotate da $q_j(t) \in \mathbb{R}^2$, con $j \in \{1, \dots, M\}$. Un gruppo di N **Agenti mobili, dinamici** (*Agents*), ad un'altezza fissata, è incaricato di monitorare continuamente questi targets. Le posizioni degli agenti sono denotate da $p_i(t) \in \mathbb{R}^2$, con $i \in \{1, \dots, N\}$.

Si presume che ogni agente abbia una zona di rilevamento limitata, modellata da un'area discoidale con un certo *raggio di copertura* $r \in \mathbb{R}^+$ centrata su se stesso.

Poiché la qualità del rilevamento generalmente degrada con la distanza dal target, il rilevamento basato sulla distanza dell'agente i sul target j è caratterizzato da una **FUNZIONE DI MISURAZIONE**:

$$E_{ij}(p_i(t), q_j(t)) = \begin{cases} M_P \left(\frac{(l_{ij}(t)-r)^2}{r^4} \right), & \text{se } l_{ij}(t) \leq r^2, \\ 0, & \text{se } l_{ij}(t) > r^2, \end{cases} \quad (1)$$

dove $l_{ij}(t) = \|p_i(t) - q_j(t)\|^2$ è la distanza, all'istante t , tra il drone i e il target j , e $M_P \in \mathbb{R}^+$ è una qualità di rilevamento di picco. Il modello di rilevamento indica che la qualità del rilevamento del target j raggiunge il picco quando l'agente i coincide con esso, e diminuisce monotonamente quando l'agente i si allontana da esso.

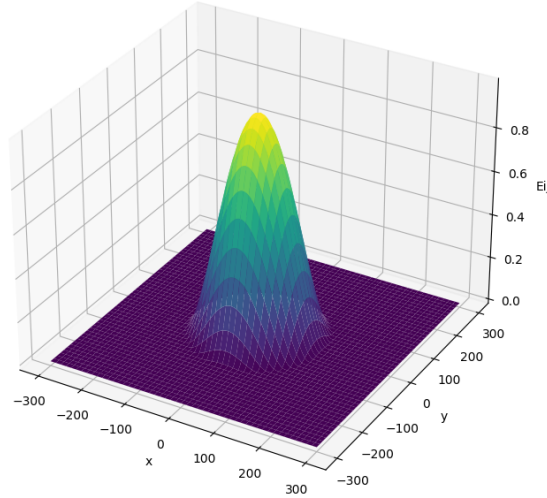


Figura 2: Funzione di misurazione, $mp=1$, $r=150$ (3).

1.3 Soluzioni proposte

In questa sottosezione viene definita la procedura matematica che stabilisce il *grado di copertura sufficiente*, menzionato nel paragrafo 1.1.

Il controllo della copertura efficace basato su *RL* (Reinforcement Learning) in questo lavoro ha l'obiettivo di guidare gli agenti mobili affinché garantiscano che ogni target nella regione di interesse sia coperto cumulativamente a un livello desiderato nel tempo (2). Per ogni target j , viene definito un *indice di copertura* $E_j(t)$, che quantifica la qualità della copertura al tempo t fornita dai vari agenti. Ogni agente contribuisce all'indice di copertura in base alla propria vicinanza al target, secondo la seguente funzione di misurazione:

$$E_j(t) = \sum_{i=1}^N E_{ij}(p_i(t), q_j(t)), \quad j \text{ fisso} \quad (2)$$

Come indicato nella sezione 1.2, la funzione di misurazione $E_{ij}(p_i(t), q_j(t))$ varia nell'intervallo $[0, M_P]$. Di conseguenza, l'indice di copertura $E_j(t)$ apparterrà all'intervallo $[0, M_P \cdot N]$.

Successivamente, viene definito un *indice di copertura totale* $E(t)$, espresso come:

$$E(t) = \sum_{j=1}^M \sigma(E_j(t)) \quad (3)$$

dove:

$$\sigma(E_j(t)) = \frac{\tanh(E_j(t) - E^*) + 1}{2} \quad (4)$$

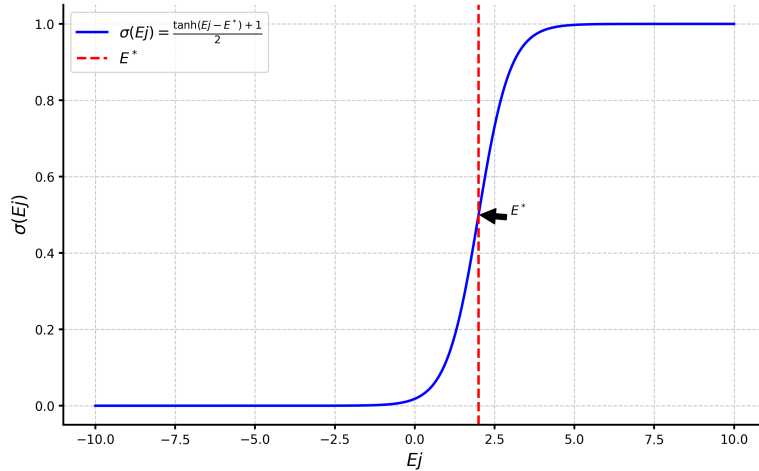


Figura 3: Funzione sigmoidale, $E^* = 2$

La definizione di queste funzioni e indici di copertura è fondamentale per raggiungere l'obiettivo del problema. La soglia E^* rappresenta un *lower bound index (LB)*, ovvero ogni indice di copertura di un target generico j al tempo t ($E_j(t)$) deve essere maggiore di questa soglia per considerare "accettabile" la copertura fornita dagli agenti. È importante notare che I PARAMETRI SONO FORTEMENTE INTERDIPENDENTI; infatti, in base a M_P , varia l'intervallo in cui è contenuta la funzione di misurazione $E_{ij}(p_i(t), q_j(t))$ e, di conseguenza, anche gli indici di copertura ($E_j(t)$). È quindi necessario settare i vari parametri caso per caso. Nella simulazione effettuata (vedi 4), sono stati fissati i seguenti parametri:

- $N = 4$. Rappresenta il numero degli agenti.
- $M = 10$. Rappresenta il numero dei target.
- $M_p = 1$. Definisce l'intervallo in cui è contenuta la funzione di misurazione e anche gli indici di copertura.
- $E^* = 2$. Definisce una soglia minima per gli indici di copertura.
- $duration = 150$. Durata in secondi delle traiettorie dei target.

Considerando questi parametri numerici, possiamo affermare che la funzione di misurazione varia tra 0 e 1, mentre gli indici di copertura variano tra 0 e 4. Impostare la soglia E^* a 2 implica che, nel complesso, desideriamo che almeno 2 agenti su 4 coprano ciascun target. Per raggiungere il valore 2, è possibile che 2 agenti siano esattamente sopra i target e abbiano quindi copertura massima, oppure che la somma delle funzioni di misurazione complessive sia pari a 2 (ogni agente copre in minima parte quel target, sommando le coperture si raggiunge la soglia).

La funzione sigmoideale è stata definita per tener conto di questo ragionamento sulla soglia. L'indice di copertura totale somma i singoli indici di copertura a cui è applicata la sigmoide, in modo che, durante il calcolo del gradiente (vedi 2), tutti i contributi degli agenti siano considerati.

Durante la simulazione, verranno generate delle posizioni iniziali degli agenti tali da avere una copertura leggermente inferiore al sufficiente nella situazione iniziale. L'obiettivo sarà quello di sviluppare un algoritmo di copertura dinamica (che verrà analizzato nella sezione 2) per creare le traiettorie degli agenti e indicare loro come muoversi, al fine di ottenere un *indice di copertura totale* finale significativamente maggiore rispetto a quello iniziale $E(duration - 1) > E(0)$.

Può succedere, come verrà analizzato in modo esaustivo nella sezione 4, che alcuni indici di copertura singoli al tempo finale $E_{ij}(duration - 1)$ siano azzerati o prossimi allo zero. Questo non rappresenta un problema, ma è un comportamento previsto dell'algoritmo di ascesa del gradiente (trattato nella sezione 2), che punta a massimizzare l'indice di copertura totale. Di conseguenza, alcuni target potrebbero non essere coperti nell'istante finale; tuttavia, ciò avvantaggia molti altri target che verranno coperti in maniera ottimale, contribuendo così ad aumentare l'indice totale.

2 Algoritmi

- 2.1 Algoritmo di coverage V1: algoritmo di ascesa del gradiente classico
- 2.2 Algoritmo V2: aggiunta di un moto Browniano
- 2.3 Algoritmo V3: aggiunta di un potenziale repulsivo

3 Implementazione degli algoritmi

3.1 Descrizione del codice (e tabelle con pseudocodice)

3.2 Librerie Python utilizzate

4 Simulazione e discussione dei risultati

5 Conclusioni

Riferimenti bibliografici

- [1] J. Cortes, S. Martinez, T. Karatas and F. Bullo, "Coverage control for mobile sensing networks," in IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 243-255, April 2004, doi: 10.1109/TRA.2004.824698. keywords: Sensor phenomena and characterization;Remotely operated vehicles;Mobile robots;Temperature sensors;Biosensors;Animals;Optimization methods;Partitioning algorithms;Prototypes;Infrared sensors, <https://ieeexplore.ieee.org/document/1284411>
- [2] S. Meng and Z. Kan, "Deep Reinforcement Learning-Based Effective Coverage Control With Connectivity Constraints," in IEEE Control Systems Letters, vol. 6, pp. 283-288, 2022, doi: 10.1109/LCSYS.2021.3070850. keywords: Sensors;Monitoring;Dynamics;Task analysis;Reinforcement learning;Vehicle dynamics;Training;Coverage control;deep reinforcement learning;multi-agent systems, <https://ieeexplore.ieee.org/document/9395182>
- [3] Hübel, Nico. (2008). Coverage Control with Information Decay in Dynamic Environments. IFAC Proceedings Volumes. 41. 4180-4185. 10.3182/20080706-5-KR-1001.00703. https://www.researchgate.net/publication/254995455_Coverage_Control_with_Information_Decay_in_Dynamic_Environments