SLIDE 1
Buongiorno a tutti. Il mio progetto di tesi tratta del CONTROLLO DI COPERTURA DINAMICO PER IL MONITORAGGIO COOPERATIVO IN TEMPO REALE.
SLIDE 2
Il controllo di copertura dinamico è un problema centrale in molti campi applicativi, tra cui la robotica, la sorveglianza e la gestione delle emergenze, è dunque un tipo di controllo cooperativo che richiede a un sistema multi-agente di monitorare dinamicamente un'area di interesse nel tempo.

(una flotta di agenti mobili, come per esempio dei droni)

~~Nel progetto è dunque necessario garantire che un'area sia controllata in modo efficace e continuo da una flotta di agenti mobili, come ad esempio dei droni~~. Questo compito diventa particolarmente complesso e interessante quando l'ambiente è dinamico e i punti di interesse (targets) sono in movimento.

Saranno dunque presenti dei target mobili, e l'obbiettivo è quello di scrivere un algoritmo di coverage che permetta di determinare le traiettorie degli agenti (ad un'altezza FISSATA) al fine di coprire al meglio i target dinamici nel tempo. Si dovrà mantenere il più alto livello possibile di copertura totale per tutta la durata del monitoraggio, adattandosi dinamicamente ai cambiamenti dell'ambiente.
L'algoritmo è basato su una strategia GREEDY, che guarda quindi l'istante attuale, senza effettuare previsioni. Inoltre, come tecnica di ottimizzazione è stata scelta l'ascesa del gradiente al fine di massimizzare una funzione obbiettivo che poi vedremo.
L'implementazione dell'algoritmo è stata fatta in tre versioni differenti, al fine di risolvere un problema di collisione presente inizialmente.
SLIDE 3
Ma com'è definito a livello matematico il problema? È importante capirlo a fondo al fine di comprendere al meglio il senso logico di tutto il progetto. Sono presenti vari parametri: ELENCO PARAMETRI LEGGENDO SLIDE….
Successivamente, come vediamo, viene definita una funzione di misurazione Eij, calcolata per ogni agente i e per ogni target j. È molto importante notare che essa è compresa tra 0 ed Mp. Essa sarà pari a 0 quando il target j è al di fuori del raggio di copertura dell'agente i, mentre sarà massima quando l'agente è precisamente sopra al target.

SLIDE 4
Si definiscono poi altre funzioni, tra cui l'indice di coperta Ej, calcolato per ogni target e compreso tra 0 e Mp*il numero di agenti N. Si fissa j e si somma tutte le

funzioni di misurazione sull'indice i. Dunque, ogni agente darà un contributo che vale massimo Mp per la copertura del target j.
Molto importante sarà poi l'indice di copertura totale, che rappresenta la funzione obbiettivo, colei che viene massimizzata dall'algoritmo di coverage. Per garantire la derivabilità e considerare il contributo di tutti gli agenti è stata implementata una soglia soft tramite una funziona sigmoidale. E* invece rappresenta un LB, si richiederà dunque che ogni indice Ej sia superiore a questa soglia, al fine di coprire in modo soddisfacente quel target.

SLIDE 5

Come già accennato, l'algoritmo si basa sulla strategia dell'ascesa del gradiente come tecnica di ottimizzazione. Si vuole massimizzare l'indice di copertura totale, per farlo, ogni agente calcola in modo distribuito il gradiente rispetto alla propria posizione. Successivamente ogni agente si muove nella direzione determinata dal gradiente. La figura infatti rappresenta proprio questo concetto. Prendendo come esempio una funziona col massimo in 0,0 si vede come, con vari steps, vado ad avvicinarmi all'obbiettivo.

SLIDE 6

In questa slide viene mostrato lo pseudocodice della prima versione dell'algoritmo. In questa versione, si utilizza esclusivamente l'ascesa del gradiente e, come detto poco fa, si calcola le derivate, e con l'istruzione in rosso si aggiorna delle posizioni degli agenti. La posizione al tempo t+1 è data dalla posizione al tempo t + il passo di salita epsilon per la componente specifica del gradiente. Il parametro epsilon è direttamente responsabile della velocità con cui gli agenti si muovono seguendo la traiettoria determinata dal gradiente della funziona obbiettivo.
Un problema emergente in questa situazione è che gli agenti, per massimizzare la copertura, possono convergere verso posizioni finali molto vicine tra loro aumentando il rischio di collisione. Questo è un problema ben noto nell'ambito dei sistemi multi-robot. Per mitigare il rischio di collisione sono state sviluppate due ulteriori versioni dell'algoritmo.

SLIDE 7

Nella seconda versione, le posizioni degli agenti vengono aggiornate non solo utilizzando il gradiente, ma anche includendo una componente stocastica derivante dal moto browniano (anche conosciuto come "la camminata del marinaio ubriaco"). Questo approccio introduce una variazione casuale ~~nelle traiettorie~~ nello spostamento degli agenti, ~~riducendo il rischio di collisione~~.
Il moto browniano è caratterizzato da una traiettoria irregolare e imprevedibile, dove ogni passo successivo è indipendente dal precedente. Eta_i(t) rappresenta la componente Browniana, la cui direziona casuale teta è distribuita uniformemente nell'intervallo 0,2pigreco mentre l'intensità del moto segue una distribuzione normale di Gauss.
Come si nota sempre dall'istruzione in rosso, prima del ciclo interno si calcola la componente che viene poi sommata per l'aggiornamento della posizione degli agenti.

SLIDE 8

Invece, nella terza versione dell'algoritmo le posizioni degli agenti vengono aggiornate includendo una componente di potenziale repulsivo. Questo approccio introduce una forza repulsiva tra gli agenti quando la distanza tra di loro è inferiore a una certa soglia δ. L'obiettivo principale è evitare collisioni tra gli agenti, migliorando la sicurezza e l'efficacia del controllo. Come si vede nell'aggiornamento delle posizioni, viene sommata una forza repulsiva.

SLIDE 9

Essa è calcolata come la somma di tutte le componenti repulsive dell'agente 'i' rispetto a tutti gli altri agenti 'k', con i diverso da k ovviamente. Ogni componente repulsiva è composta dal primo termine che indica la direzione di allontanamento dall'agente k e il secondo che indica l'intensità della forza repulsiva, di cui ne possiamo vedere il grafico. Com'è giusto che sia, l'intensità decresce all'aumentare della distanza d tra i due agenti, e quando essa è inferiore a delta, allora sarà alta e darà il suo contributo nell'allontanamento degli agenti.

E' importante però andare a verificare che questa aggiunta della componente repulsiva non vada a sovrastare il comportamento principale dell'algoritmo, che rispetta l'ascesa del gradiente.

SLIDE 10

Per quanto riguarda l'implementazione degli algoritmi, è stata fatta in Python, in particolare sono state utilizzate le librerie che vediamo. E' stato fondamentale l'utilizzo di Pytorch, una libreria open-source per il training dei modelli di machine learning, che ha permesso un calcolo davvero preciso del gradiente, ~~dal momento che il metodo alle differenze finite presentava oscillazioni troppo brusche del gradiente~~. Infine, prima dell'esecuzione degli algoritmi sono state effettuate varie operazioni preliminari, tra cui il setting dei parametri iniziali, l'estrazione delle traiettorie dei targets (unico dato in input del progetto), sono state generate le posizioni iniziali degli agenti e sono stati calcolati gli indici di copertura all'istante iniziale.

SLIDE 11

Come è possibile notare dalla figura, il dataset è composto da delle traiettorie in uno spazio libero non strutturato, le quali sono similari a delle traiettorie di imbarcazioni, non seguono una strada come se fossero delle macchine ad esempio.

È stato preso un numero di agenti pari a 4, meno della metà del numero di target (10, SELEZIONATI IN MODO COMPLETAMENTE CASUALE DAL DATASET COMPLETO). La durata delle traiettorie dei target è di 150 secondi. Molto importante notare la scelta di Mp, pari a 1. In questo modo ogni agente da un contributo di massimo un'unità nella copertura di un target, quindi impostare la soglia E* a 2 vuol dire che in media, richiediamo che almeno due agenti coprano un determinato target, o che la somma dei contributi di copertura degli agenti sia almeno pari a 2. Le simulazioni sono poi state effettuate con una richiesta di coverage più forte (LB=2) e meno restrittiva (LB=1).

SLIDE 12

Passando alle simulazioni, partendo dal caso con E*=2, possiamo vedere i grafici delle traiettorie sia di agenti che di target in tutte e tre le versioni dell'algoritmo. I cerchi indicano i punti di partenza, mentre i quadrati o le x indicano le posizioni finali.
Come notiamo dai grafici, gli agenti vanno a muoversi al fine di coprire al meglio i target mobili. Nella prima immagine gli agenti vanno a collidere nelle loro posizioni finali, nella seconda versione il problema non viene risolto in modo soddisfacente mentre grazie al potenziale repulsivo si nota come, nella terza immagine, essi sono sicuramente a una distanza superiore di quella di sicurezza delta.

SLIDE 13

Inoltre, in tutte e tre le versioni, l'indice di copertura totale, ovvero la funzione obbiettivo da massimizzare, cresce di molto rispetto al valore iniziale e rimane ad un valore soddisfacente. Da notare che il valore dell'indice all'istante finale nella terza versione dell'algoritmo è pressoché uguale agli altri. Questo vuol dire che l'aggiunta del potenziale repulsivo va ad eliminare totalmente il problema della collisione, mantenendo comunque una copertura ottimale.

SLIDE 14

Con questa richiesta di copertura, come si nota sia dalla tabella che dal grafico della situazione finale, gli indici singoli di copertura dei target con j=0 e j=8 sono praticamente azzerati. Questo non è un problema, semplicemente l'algoritmo, al fine di raggiungere l'obbiettivo della massimizzazione dell'indice globale con questo tipo di parametri, va a sacrificare alcuni target in favore di coprirne con una qualità eccelsa altri. Infatti, gli altri indici sono molto superiori alla soglia di 2.
CON LA SOGLIA IMPOSTATA A 2, RICHIEDAMO DUNQUE CHE ALMENO DUE AGENTI VADANO A COPRIRE UN TARGET O CHE LA SOMMA DEI CONTRIBUTI SIA UGUALE A 2. QUESTA RICHIESTA PUO' ESSERE ANCHE AMMORBIDITA, IN MODO DA ANALIZZARE I CAMBIAMENTI SULLA COPERTURA.

SLIDE 15

La simulazione è stata dunque eseguita anche tramite una richiesta di coverage meno restrittiva, impostando dunque la soglia E* pari a 1.
Come è possibile notare, non è presente il problema della collisione; gli agenti andranno ad addensarsi meno rispetto al caso precedente, in quanto appunto un target richiede di essere "coperto" in modo meno qualitativo.

SLIDE 16

Dunque, l'andamento dell'indice di copertura totale cambia, l'incremento è minore, ma comunque il valore finale della funzione obbiettivo ha un valore superiore, di circa 7, contro i 6 del caso dove impostavamo la soglia a 2. Questo miglioramento è dovuto al cambio del parametro di LB.

SLIDE 17

Dunque, si intuisce come I PARAMETRI SIANO FORTEMENTE INTERDIPENDENTI. E' fondamentale comprendere il contesto specifico per trovare un compromesso ottimale tra il numero di target, il numero di agenti e la copertura

richiesta.
Di conseguenza, anche gli indici di copertura singoli hanno valori migliori, sono quasi tutti nettamente sopra il valore uno, e anche quel target che prima era completamente scoperto, viene comunque visto in maniera minima. Qua si nota anche che all'istante iniziale era presente lo stesso un coverage ottimo, che viene migliorato all'istante finale. Ecco perché l'incremento della funzione obbiettivo è meno drastico.

SLIDE 18 CONCLUSIVA

Per concludere, è possibile affermare che, tramite un'accurata simulazione e un'analisi degli indici di copertura, gli algoritmi implementati aumentano e garantiscono nel tempo un livello di coverage soddisfacente dei targets mobili.
In particolare:

1. La prima versione dell'algoritmo ha mostrato un notevole miglioramento dell'indice di copertura totale, ma ha presentato problemi di collisione ~~tra gli agenti nelle posizioni finali~~.
2. Nella seconda versione, l'introduzione del moto browniano ha mitigato parzialmente il problema delle collisioni, sebbene non in modo completamente soddisfacente.
3. Nell'ultima versione, l'inclusione del potenziale repulsivo ha risolto efficacemente il problema delle collisioni senza compromettere l'aumento dell'indice di copertura totale, garantendo una distribuzione spaziale ottimale degli agenti.

Inoltre, si è notato come la flessibilità nella richiesta di copertura, la quale conduce a una gestione più efficiente delle risorse degli agenti, migliora ulteriormente la qualità del monitoraggio.
Per quanto riguarda alcuni possibili sviluppi di questo progetto è possibile:

1. In primis, è possibile andare oltre all'approccio greedy, integrando la predizione della posizione dei target OPPURE utilizzare tecniche avanzate come il RL, DRL, MADRL o approcci model-based che apprendono dall'ambiente la strategia ottimale di copertura.
2. Un secondo sviluppo potrebbe riguardare la modifica dell'altitudine degli agenti. Un aumento permette agli agenti di coprire un'area più vasta, ma a scapito di una risoluzione inferiore. La sfida risiede dunque nel trovare un compromesso tra questi due fattori.
3. Per concludere, sarebbe possibile eseguire un'esplorazione dell'area di interesse. Nel progetto si presume di conoscere le traiettorie dei target; tuttavia, in scenari reali queste informazioni potrebbero non essere disponibili. Una ricognizione aiuterebbe dunque a mappare l'area e rilevare la presenza di intrusi od oggetti non previsti.