

UNIVERSITA' LA SAPIENZA

DIPARTIMENTO DI INFORMATICA

AUTONOMOUS NETWORKING

Report Primo Homework

Author:

Giordano DIONISI

1834919

Mattia LISI 1709782

Michele SPINA 1711821

Supervisor:

Prof. Andrea COLETTA

Prof.ssa Gaia MASELLI

18 novembre 2021

DIPARTIMENTO
DI INFORMATICA



SAPIENZA
UNIVERSITÀ DI ROMA

Indice

1	Let's Start!	2
1.1	Algoritmo BR ed AIB	2
1.2	Algoritmo RND_RL ed RND	3
1.3	Algoritmo CLO ed UCB	4
1.4	Algoritmo GEOUP	5
1.5	Algoritmo EGN	6
1.6	Algoritmo OIV	7
1.7	Algoritmo EGSOLOID	8
1.8	Algoritmo AI_BEST	8
1.9	Algoritmo AISG	9
2	Precisazioni	10
2.1	La Funzione <i>varepsilon</i>	10
2.2	Reward	10
2.3	Variabili Locali vs Variabili Globali	10
2.4	Per Eseguire	11
3	Sezioni implementate per componenti	11
4	Conclusione	11

1 Let's Start!

Obiettivo: trovare un algoritmo efficace nella comunicazione di pacchetti tra droni nella configurazione data.

Prefazione: Vari plots sono stati realizzati con i possibili algoritmi. Ogni plot raffigura sia l'algoritmo target (Georouting) sia il più banale (None Algorithm).

- Il **Georouting** sceglie il nodo più vicino al depot;
- Il **None**, non avvengono scambi pacchetti.

1.1 Algoritmo BR ed AIB

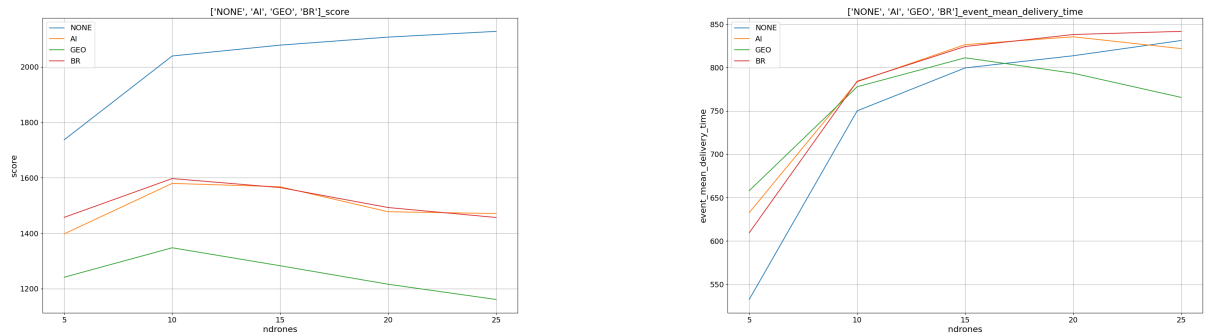


Figura 1: AI = AIB

- **BR**¹: Si invia il pacchetto al vicino con minore batteria
- **AIB**²: Con probabilità:
 - $1 - \varepsilon$ fa "*exploiting*", cioè sfrutta il RL³ → Seleziona il nodo con Q-Value maggiore.
 - A parità Q-Value privilegia il drone con minore batteria;

¹Battery Routing

²AIRouting Battery

³Reinforcement Learning

- ε sceglie il nodo con minore batteria.

Entrambi non portano buoni risultati, poiché le informazione sulla carica non sono sufficienti.

1.2 Algoritmo RND_RL ed RND

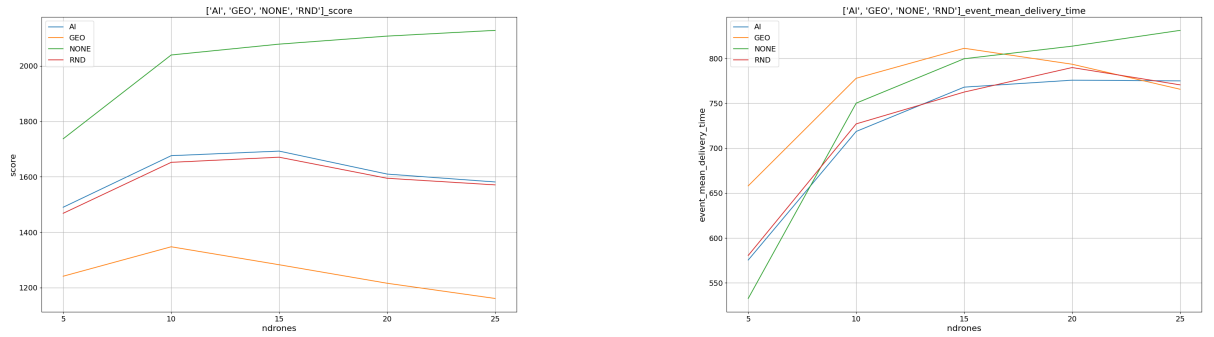


Figura 2: AI = RND_RL

- **RND:** Sceglie randomicamente uno dei possibili nodi cui mandare il messaggio tra i vicini (incluso se stesso).
- **RND_RL:** Tra le alternative, sceglie il nodo in maniera casuale a seconda del Q-Set⁴ locale per ciascun drone.

Sono poco efficienti, infatti ottengono risultati imparagonabili rispetto al GeoRouting.

Entrambi racchiudono la stessa idea fondata su basi diverse: randomicità
→ Cio' spiega l'andamento di entrambi molto simile, a livello di convergenza.

⁴Contiene i Q-Values

1.3 Algoritmo CLO ed UCB

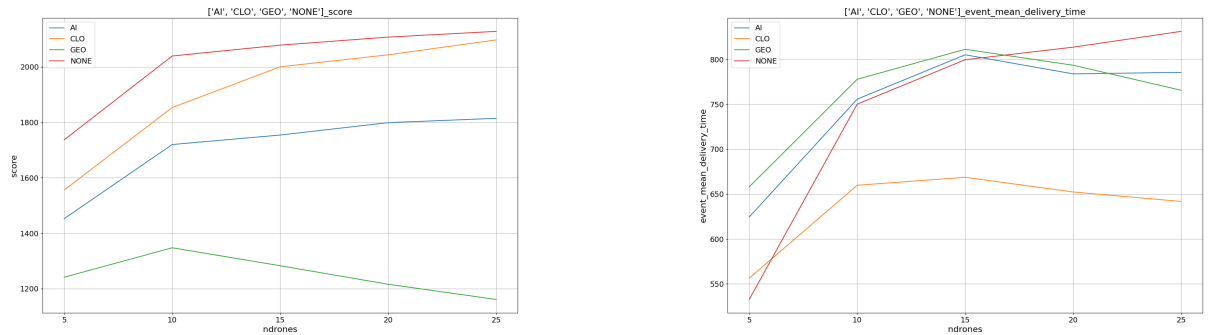


Figura 3: AI = UCB

- **CLO⁵**: L'azione e' di passare il pacchetto al nodo piu' vicino al nodo stesso;
- **UCB⁶**: Sceglie il nodo con maggior potenziale, cioe' il massimo del suo intervallo di incertezza e' il maggiore le possibilita'.

CLO ha risultati fortemente negativi, infatti l'eccessivo passaggio di pacchetti porta a prestazione peggiori del None Algorithm.

L'UCB, con pochi droni, ha prestazioni simili al GeoRouting → Per definizione non e' adatto a problemi non stazionari, come quello in analisi⁷, quindi gia' in partenza e' un algoritmo inadeguato e lontano dall'approccio target (GeoRouting).

⁵Closest To Me

⁶Upper Confidence Bound

⁷Restringe la soglia di confidenza, mentre questo problema comprende anche il fatto che tale soglia puo' riallargarsi

1.4 Algoritmo GEOUP

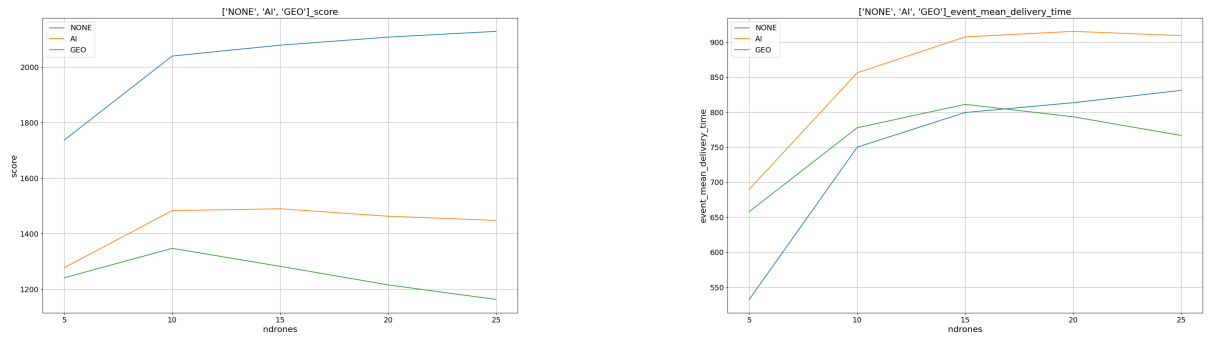


Figura 4: AI = AISG

Il GEOUP⁸ Effettua una stima della posizione del drone vicino e poi seleziona come nodo colui con minore distanza tra la propria traiettoria (*il segmento che unisce la sua posizione attuale con la sua destinazione*) ed il depot.

E' simile al GeoRouting, quindi l'andamento dei due e' analogo ma i risultati del GEOUP sono, sperimentalmente, leggermente peggiori rispetto al GeoRouting.

⁸Geographical Upgraded

1.5 Algoritmo EGN

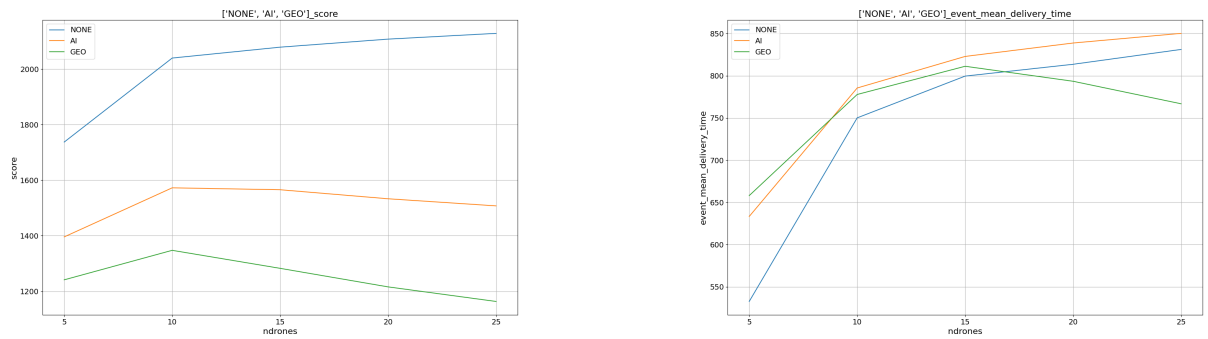


Figura 5: AI = EGN

L'EGN⁹ con una probabilità':

- $1-\varepsilon$ adotta l'approccio di RL scegliendo il massimo Q-Value \rightarrow A parità di Q-Values sceglie il nodo che passa più vicino al depot (con distanza minore tra traiettoria e depot);
- ε sceglie randomicamente il vicino cui passare il pacchetto, "*exploration*".

L'EGN ha prestazioni discrete ma risulta un punto di partenza per altri miglioramenti algoritmici.

⁹Epsilon-Greedy Normal

1.6 Algoritmo OIV

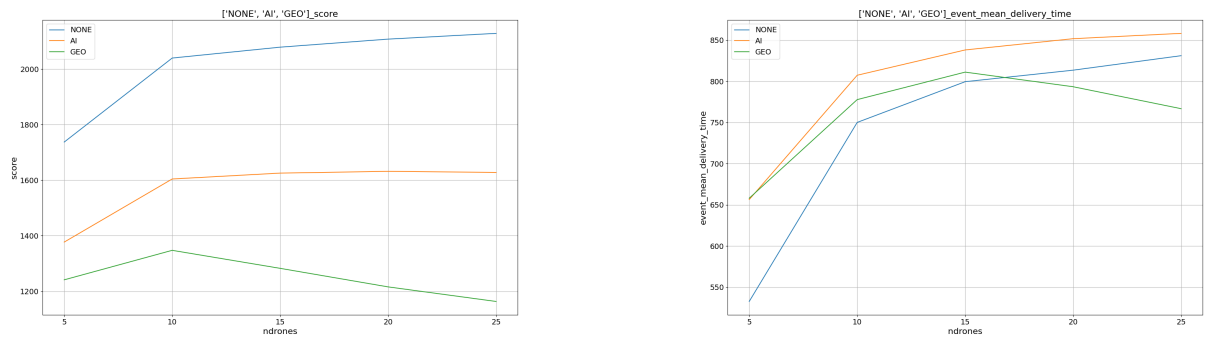


Figura 6: AI = OIV

Funziona come l'EGN, ma inizialmente le Reward hanno un valore maggiore. Cio' favorisce una maggiore iniziale esplorazione, per poi avere una *exploitation* piu' massiccia successivamente.

Le prestazioni tendono ad essere costanti (come nell'approccio None) → All'aumentare dei droni non ci sono diverse performance. Cio' infica che non e' un algoritmo ottimale.

Si e' cambiato ε e la Reward per avere migliori performance, ma tendenzialmente l'andamento rimane uguale (molto lineare) e per questo si e' deciso di non procedere.

1.7 Algoritmo EGSOLOID

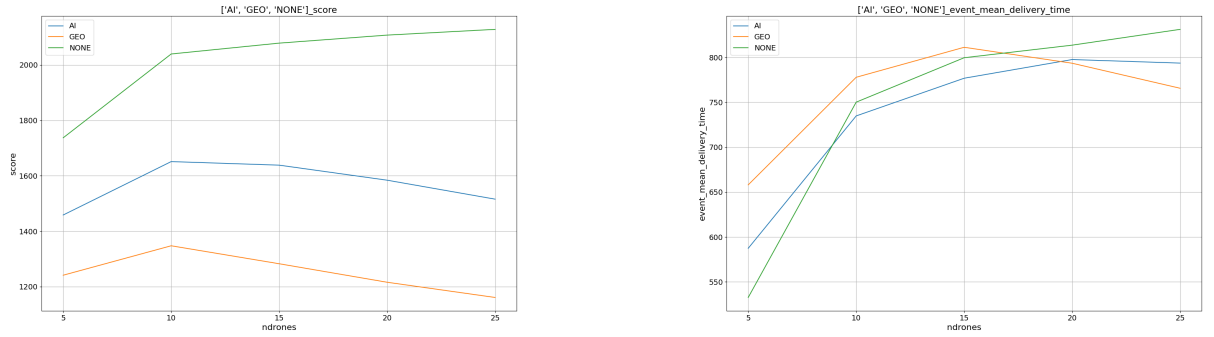


Figura 7: AI = EGSOLOID

Identico all'algoritmo EGN ma il Q-Value e le Rewards si attribuiscono all'identificativo del drone considerato, rimuovendo la destinazione dalle chiavi.

La perdita di quest'ultima informazione ha comportato un calo delle prestazioni, pertanto in tutti gli altri algoritmi di RL realizzati utilizzano come chiavi sia l'id che la destinazione dei droni.

1.8 Algoritmo AI_BEST

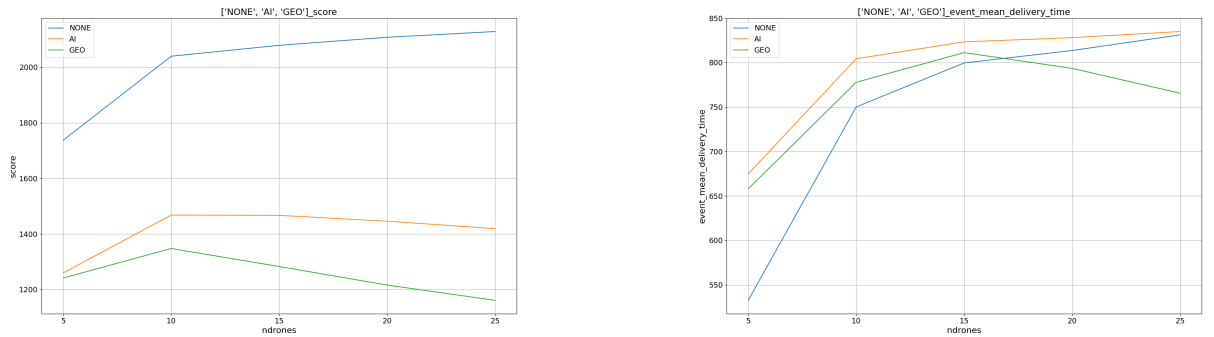


Figura 8: AI = AI_BEST

AI_BEST applica i principi basilari del RL privilegiando il drone con la traiettoria piu' vicina al depot, anziche' effettuare l'*esplorazione* randomicamente. Inoltre se piu' droni hanno uguale Q-Value, AI_BEST sceglie tra loro quello con distanza minore tra la propria traiettoria ed il depot.

Questo algoritmo, con pochi droni, ha dato inizialmente eccellenti risultati; ma all'aumentare dei droni le sue prestazioni non hanno un miglioramento sufficiente ad eguagliare il GeoRouting. Risulta però essere un valido algoritmo.

1.9 Algoritmo AISG

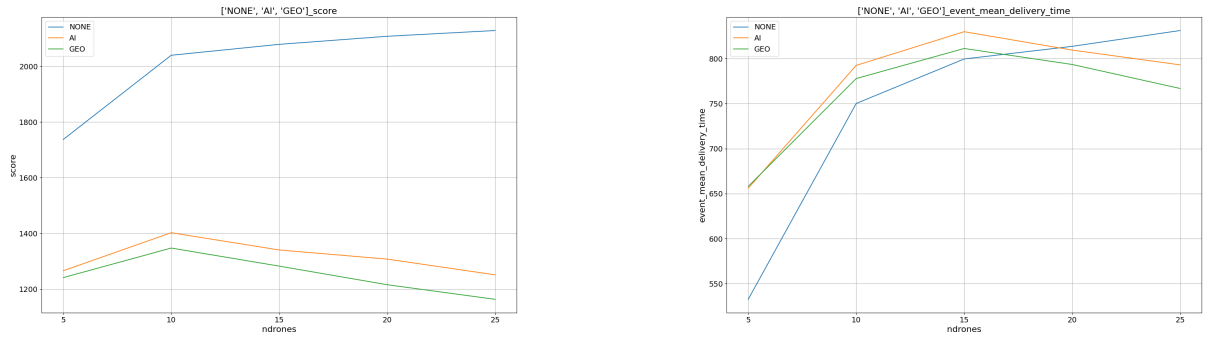


Figura 9: AI = AISG

Similmente all'AI_BEST, applicando i principi del GeoRouting anziché del GEOUP, con probabilita':

- $1 - \varepsilon$ prende il drone con il maggiore Q-Value \rightarrow a parita' di Q-Values sceglie il drone piu' vicino al depot;
- ε prende il nodo piu' vicino al depot.

AISG ha ottime performance \rightarrow Ha performance simili al GeoRouting, con un tempo di consegna leggermente maggiore. Pertanto risulta algoritmo realizzato tramite Reinforcement Learning.

2 Precisazioni

2.1 La Funzione *varepsilon*

AIB, AISG ed AI_BEST utilizzano una funzione ε definita in un sotto-intervallo di $[0, 1]$ inversamente proporzionale alla mole di dati utili al RL e direttamente proporzionale al numero di possibili scelte. Questa scelta ha portato ad un miglioramento dello score negli algoritmi, rafforzando la probabilità di utilizzo dei Q-Values, quando più utili, e utilizzando maggiormente l'*esplorazione* quando più necessari, poiché le informazioni a disposizione sono più scarse. Al fine di sfruttare maggiormente il RL è stata costruita una funzione il cui andamento sia schiacciato verso il minimo di essa.

2.2 Reward

La Reward è una funzione in -1 *union* $[1,2]$:

- In caso di mancato arrivo del pacchetto in tempo, Reward = -1;
- In caso di arrivo del pacchetto si considera il delay¹⁰:
 - Più è elevato e più la Reward è vicina ad 1
 - Più è piccolo e più la Reward è vicina a 2

La Reward va incrementata/decrementata allo specifico Q-Value.

Si è scelta, dopo tentativi, questa Reward perché considera anche il delay del pacchetto e perché, sperimentalmente, fornisce risultati migliori, considerando tale delay.

Per Realisticità la Reward viene attribuita a tutti i nodi che partecipano alla consegna del pacchetto, non solo al nodo più vicino alla partenza → Infatti ogni nodo è responsabile dell'invio del pacchetto e ciò comporta un training più robusto ed efficace.

2.3 Variabili Locali vs Variabili Globali

Inizialmente il Q-Set era globale → Tutti i droni accedevano e modificavano il Q-Set per i Q-Values, tramite rete cellulare.

¹⁰E' inversamente proporzionale

Ora il Q-Set e' locale, perche' quest'assunzione e' forzata \rightarrow Ogni drone ha il suo Q-Set dove memorizzare i Q-Values e sul quale fare training. La situazione e' piu' realistica, perche' ogni drone puo' considerare diversamente gli altri possibili droni.

2.4 Per Eseguire

Per eseguire il .sh occorre richiamare un solo algoritmo di RL alla volta. In allegato vi e' anche il file .config modificato.

3 Sezioni implementate per componenti

Da sottolineare che il progetto e' stato realizzato in gruppo, quindi ognuno ha contribuito ugualmente alla sua realizzazione (le decisioni sono state prese di comune accordo).

Nello specifico le differenze sono:

- **Giordano:** Si e' occupato di costruire la maggior parte degli algoritmi di RL e non solo a livello base, per poter poi essere migliorati dagli altri componenti del gruppo. Ha concettualmente pensato alle varie possibili idee implementative ed ha, infine, provveduto principalmente alla stesura del Report.
- **Mattia:** Ha provveduto alla realizzazione e ideazione dell'algoritmo di GEOUP e la revisione del report.
- **Michele:** Si e' concentrato principalmente alla ideazione e realizzazione di BI e GEOUP e dell'applicazione di tali principi, e quelli del Georouting, per migliorare il RL negli algoritmi AI_BEST, AISG e AIB. Ha svolto la ricerca e definizione di una funzione epsilon efficace e piu' efficiente, e la revisione del report.

I

4 Conclusione

Il miglior algoritmo realizzato e' AISG. Usa un'idea simile ad AI_BEST ed e' il migliore tra gli algoritmi realizzati. E' abbastanza vicino al GeoRouting per performance e logicamente ha grande potenziale prestazionale.