Relazione su
Controllo formazione droni

A. Bettoni (1998044),
A. Coppola (2003964),
S. Di Cesare (1938649)

June 25, 2024

Contents

1 Descrizione Generale

Problema: Si vuole progettare un sistema di controllo di n droni che sorvegli una data area rettangolare. Il sistema deve garantire che, con i droni forniti, ogni punto dell'area venga sorvegliato il più frequentemente possibile.

I droni partiranno dalla torre di controllo che si trova al centro dell'area da sorvegliare. Ogni drone ha un'autonomia limitata (misurata in minuti in volo) e una velocità massima. La torre dovrà gestire gli spostamenti di ogni drone facendo in modo che tornino alla torre prima che la batteria sia del tutto esaurita. Quando i droni si trovano nella torre di controllo sono considerati in carica e il tempo di ricarica può variare da drone a drone.

Idea di soluzione: Abbiamo scelto di modellare il problema con un sistema che agisce sulla torre di controllo e ricarica che a sua volta controlla i droni. Dunque la torre invia ad ogni drone le coordinate del punto sull'area da raggiunge secondo un sistema di volo a "tappe". Una volta che il drone arriva alla posizione ricevuta lo comunica alla torre che risponde con la posizione successiva, delineando così un percorso. Quando il drone è scarico lo comunica alla torre e si direziona alla torre di controllo, per essere ricaricato.

Sta al sistema quindi il compito di calcolare, mentre i droni sono in volo, il percorso migliore per far sì che ogni punto dell'area venga sorvegliato il più frequentemente possibile, tenendo conto dei punti visitati, dei droni in volo, e dei punti che visitano prima di scaricarsi.

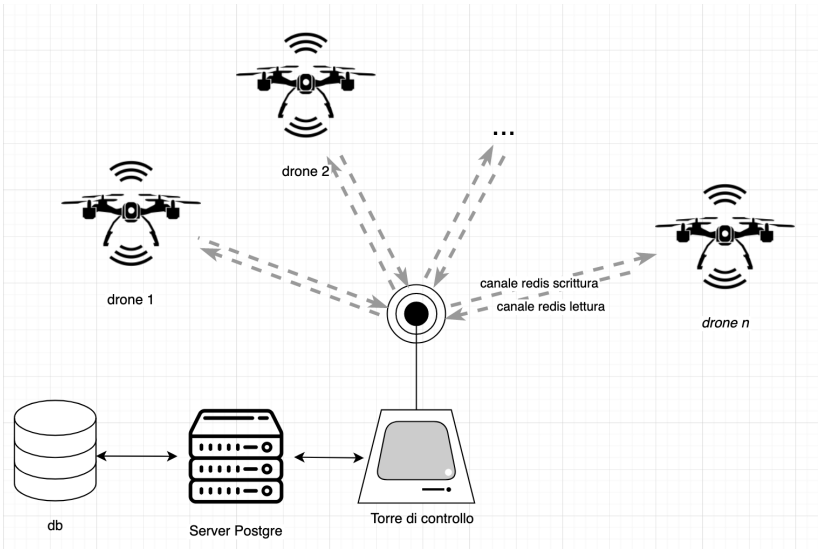

Figure 1: Architettura fisica del sistema

2 Analisi del software

2.1 Requisiti utente

Per funzionare, il sistema ha bisogno dei seguenti requisiti:

- dimensione dell'area
  dato che l'area verrà divisa in blocchi, per ogni blocco è di interesse:
  - limite alto superiore
  - limite basso inferiore
  - punto di partenza (o ultimo punto visitato)
  - Ogni punto dell'area è identificato dalla sua posizione in una matrice di punti e di quei punti ci interessa solo:
    * il tempo trascorso dall'ultima visita.
- numero di droni
  per ogni drone è di interesse:
  - id: identificativo univoco seriale
  - Stato : tra i stati precedentemente elencati
  - posizione
  - carica residua (in minuti)
  - carica massima (in minuti)
  - tempo di ricarica
  - last_update : tempo trascorso dall'ultimo update

2.2 Requisiti di sistema

Il sistema deve rispettare i seguenti requisiti:

1. Area:

   l1bel=1.0 I blocchi sono tutti tra loro disgiunti (non esiste un punto dell'area che si trova in più di un blocco)

   l2bel=2.0 L'unione dei blocchi copre tutti e solo i punti dell'area (un punto appartiene all'area se e solo se esiste un blocco che lo contiene)

   l3bel=3.0 Un blocco può essere assegnato al più ad un drone

2. Drone:

   l1bel=1.0 Un drone a cui è stato assegnato un blocco può trovarsi al di fuori di esso se e solo se:

   i. il drone è partito dalla torre e si sta posizionando verso il punto di partenza del blocco

   ii. il drone è scarico e sta tornando alla torre

   iii. il drone ha visitato tutto il blocco e si sta posizionando verso il punto di partenza di un'altro blocco

   l2bel=2.0 Un drone può volare verso un punto diverso da quello della torre se e solo se il suo tempo di volo residuo è maggiore del tempo che serve al drone per andare dalla sua posizione attuale a quella della torre

3. Torre:

   l1bel=1.0 La torre non può terminare il suo processo se esiste un drone connesso che non è tornato nella torre di controllo.

   l2bel=2.0 Tutti i punti che la torre invia ad un drone devono trovarsi all'interno dell'area assegnata al drone

4. Requisiti non-funzionali:

   l1bel=1.0 ogni punto dell'area deve essere visitato il più frequentemente possibile

   l2bel=2.0 Il tempo di risposta della torre per ogni drone deve essere ragionevolmente basso per fare in modo che i droni rimangano fermi sul posto il meno possibile

2.3 Diagrams

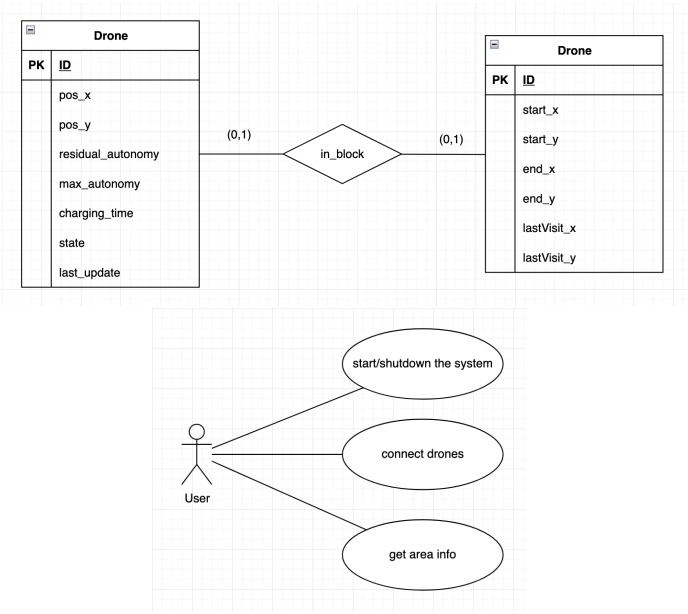

Figure 2: schema ER e UML

2.4 Algoritmo

Contando di poter ottenere per ogni drone, una tupla contenente almeno {*Id, Posizione, Stato, Blocco*} di tutti i droni che si sono connessi alla torre e le dimensioni dell'area da sorvegliare, la torre inizia a seguire il seguente algoritmo. Il problema può essere gestito seguendo i seguenti passaggi:

1. La torre di controllo conta i droni che si sono connessi e divide l'area da sorvegliare in N blocchi tutti della stessa misura secondo una specifica funzione che fa in modo di avere più blocchi che droni.

2. La torre assegna ad ogni drone *"pronto a partire"* un blocco disponibile.

3. Ogni drone si dirige al punto di partenza del blocco assegnatogli. Non appena lo raggiunge lo comunica alla torre.

4. Quindi la torre azzera il tempo di visita dell'area 20×20 centrata nella posizione del drone e gli invia il prossimo punto da raggiungere per scansionare tutto il blocco.

5. Quando il drone ha visitato tutto il blocco gli viene assegnato il blocco disponibile che contiene il punto non visitato da più tempo. Il loop continua dal punto 3.

6. Quando un drone ha carica appena sufficiente per tornare alla torre di controllo parte la procedura di return. Segnala alla torre che sta tornando comunica quindi l'ultimo punto visitato e si dirige alla torre.

7. Quando il drone arriva alla torre di controllo inizia a ricaricarsi. La torre lo marchera come *"pronto a partire"* non appena la batteria sarà carica.

8. Se la torre non riceve messaggi da un drone in volo per troppo tempo questo verrà segnato come *"morto"*.

Questo è l'idea principale per far si che tutti i punti vengano visitati il più frequentemente possibile. Questo ciclo viene ripetuto finche l'utente non interrompe il programma o tutti i droni "muoiono".

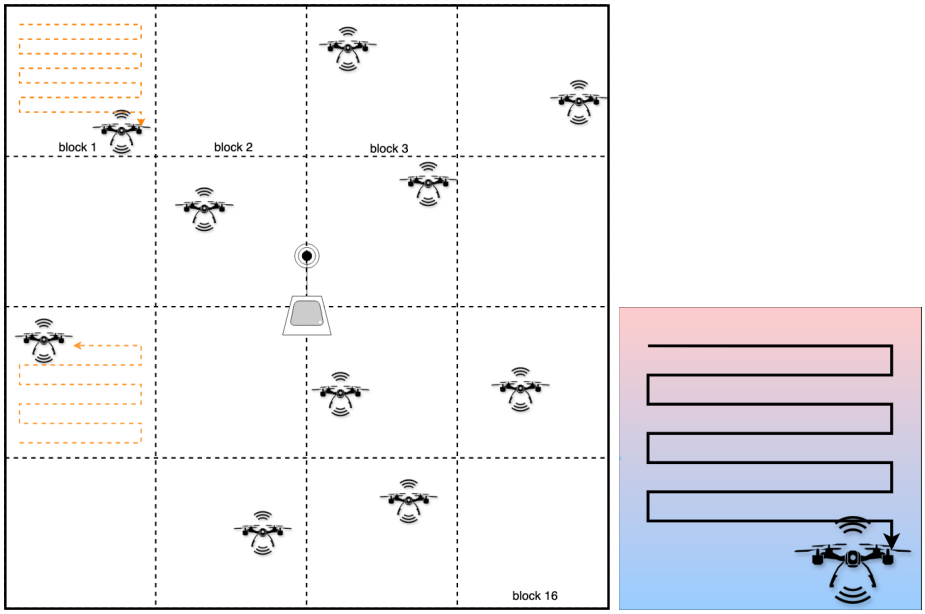

Figure 3: Esempio di area suddivisa in blocchi. Nel dettaglio a destra oltre al percorso seguito dal drone, viene evidenziata come più "calda" l'area visitata da meno tempo e più "fredda" l'area appena vista.

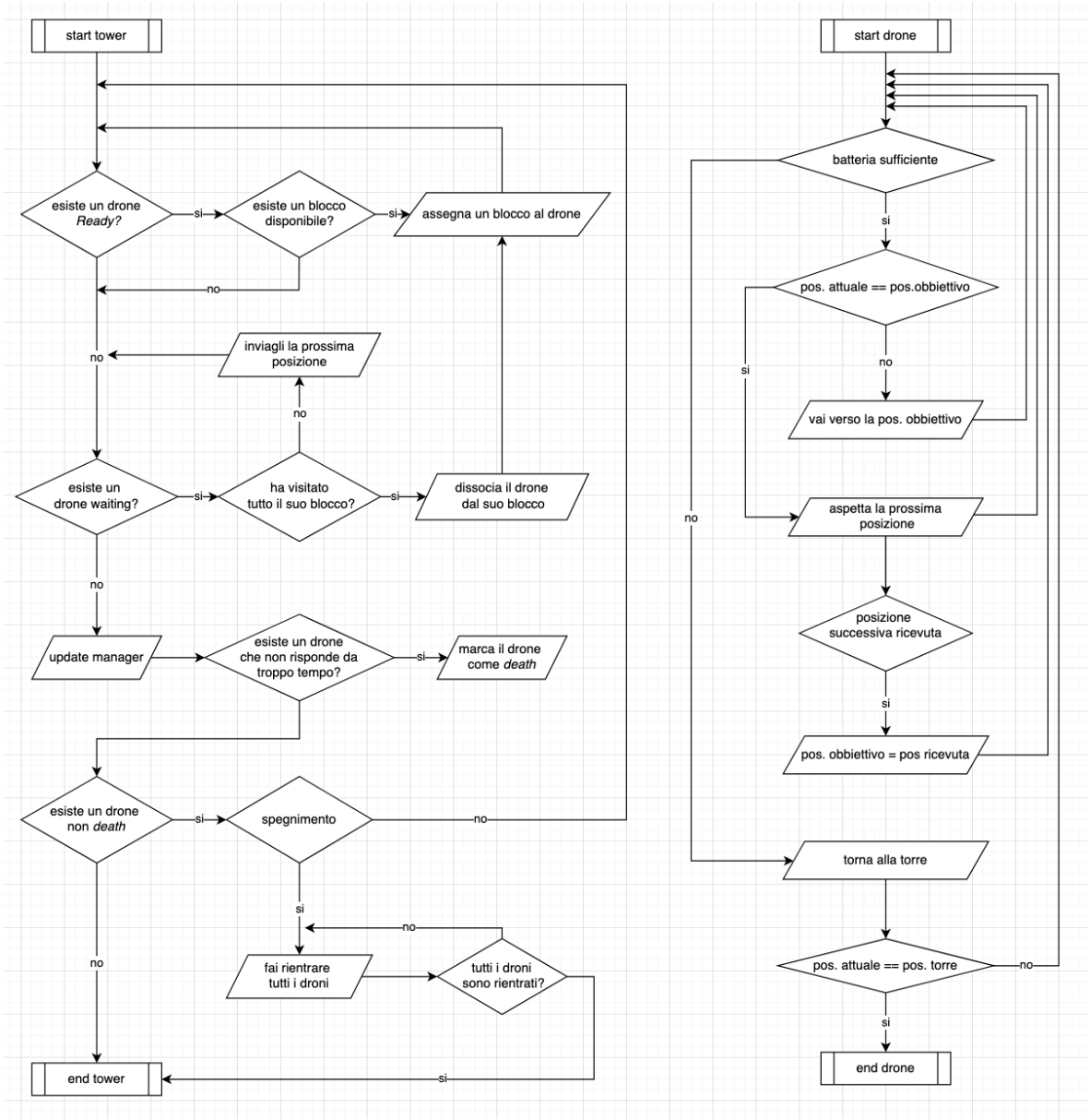

Figure 4: FlowChart della torre e del drone. Nota bene, nel FlowChart viene assunto che il drone e la torre siano già connessi

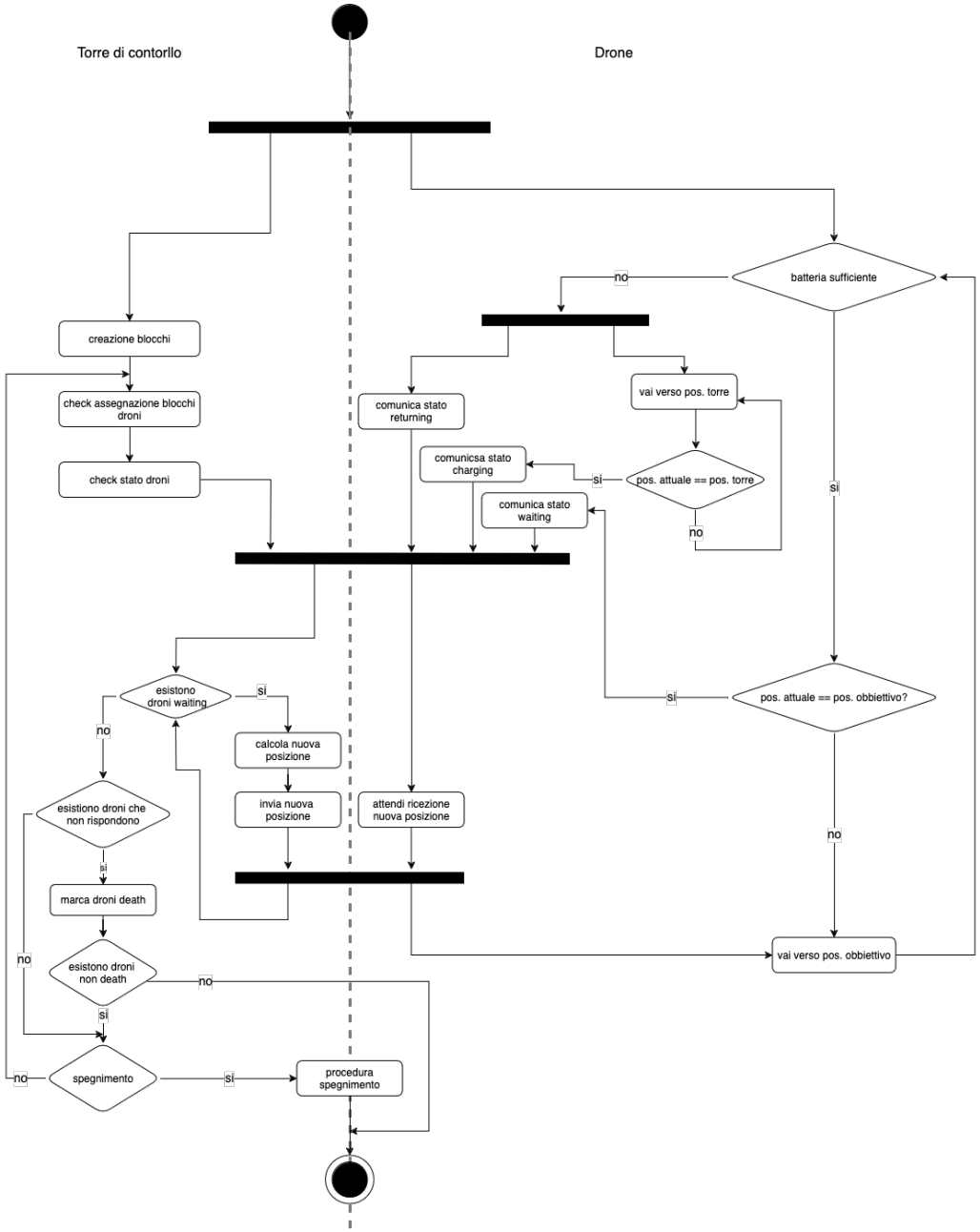

Figure 5: Activity diagram del sistema di interazione e decisione tra torre e drone, nel diagramma sono mostrate due barriere in cui drone e torre si sincronizzano per scambiarsi i messaggi principali mentre è lasciato implicito l'handler message di entrambi per messaggi di check.

2.4.1 State diagram

Per decidere l'azione da assegnare ad ogni drone la torre deve riuscire a distinguere la condizione di ogni drone. Definiamo quindi 6 stati in cui ogni drone può trovarsi:

1. **CHARGING:** il drone si trova alla base, e sta ricaricando la sua batteria
2. **READY:** il drone è carico e si trova nella torre di controllo (è pronto a partire)
3. **WAITING:** il drone sta aspettando che la torre gli invii la prossima coordinata
4. **MONITORING:** il drone sta scansionando l'area che gli è stata assegnata seguendo i punti inviati dalla torre
5. **RETURNING:** la carica del drone è sufficiente solo per il suo rientro (il drone sta rientrando)
6. **DEAD:** la batteria del drone si è scaricata prima che questo potesse rientrare oppure la torre di controllo non riesce più a contattarlo (fault).

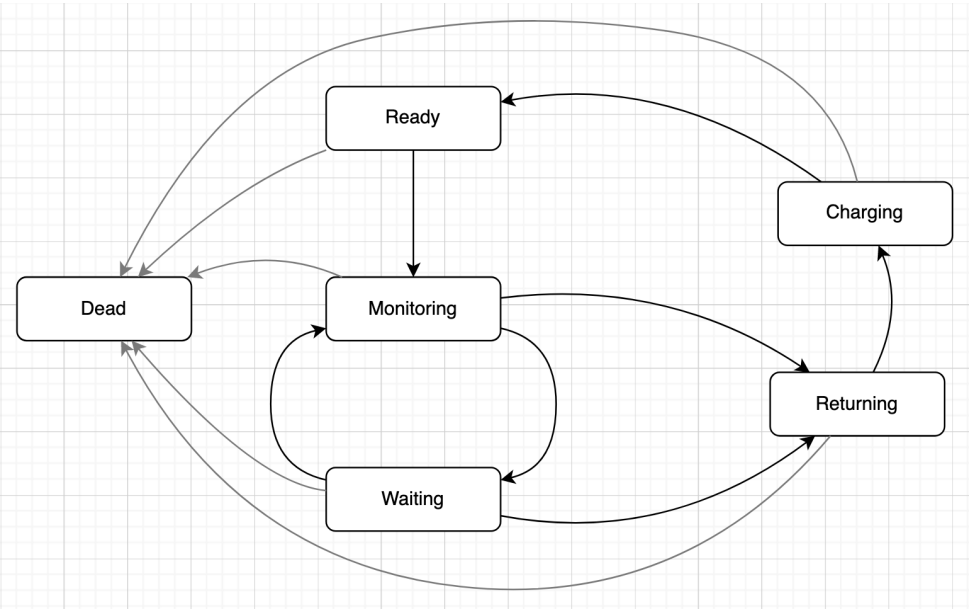

Figure 6: State diagram

2.4.2 Scelte progettuali

Divisione dei blocchi Dato che un punto si considera visitato se si trova nel raggio di 10 metri da un drone, possiamo discretizzare l'area rappresentandola come una matrice di case quadrate di dimensione $10\sqrt{(2)} \times 10\sqrt{(2)}$

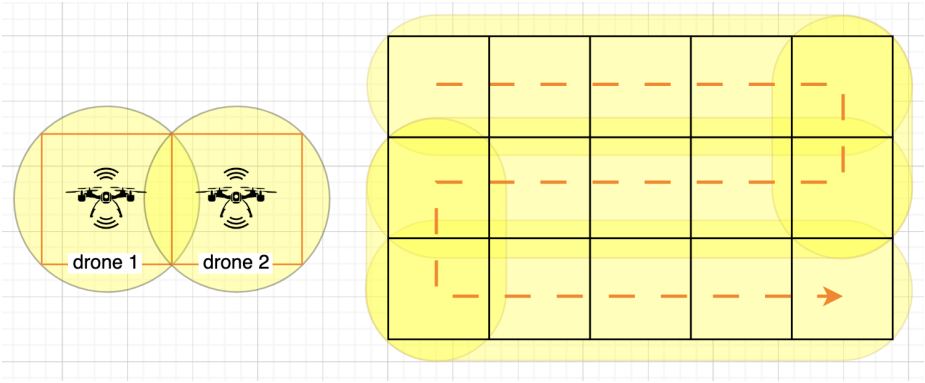

Figure 7: L'area divisa in caselle della dimensione del quadrato inscritto nel cerchio di raggio 10

Abbiamo scelto di raggruppare le caselle in blocchi tutti uguale area per garantire equità nelle assegnazioni. Data la variabilità delle dimensioni dell'area e del numero di droni in fase iniziale, non è sempre garantito che il numero di droni divida il numero caselle in un valore utile per costruire una matrice di blocchi, utilizziamo un valore di blocchi maggiore per garantirne la perfetta di visibilità in sottoaree uguali. Questo permette a tutti i droni *READY* di volare contemporaneamente all'inizio del processo di monitoraggio e quindi ridurre il più possibile il tempo di visita medio dell'intera area. Anche se creare più blocchi ne riduce le dimensioni, questo non influisce sulle prestazioni del sistema in quanto ogni drone quando termina il ciclo di visita del suo blocco inizia a visitarne un altro e continua finché la batteria lo permette.

Assegnamento dei blocchi Per minimizzare il tempo di visita massimo e medio abbiamo sperimentato due criteri di scelta:

1. **FirstAverageTime**: Assegnare ad un drone *free* il blocco con tempo medio trascorso dall'ultima visita più alto (la media viene eseguita su tutte le caselle nel blocco).
2. **FirstMaxTime**: Assegnare ad un drone *free* il blocco che contiene la casella visitata meno di recente.

Attualmente il programma utilizza il secondo approccio (**FirstMaxTime**). Se valutiamo il criterio di assegnamento basandoci su tempo di visita medio e tempo di visita massimo raggiunto durante una simulazione, possiamo osservare che a parità di area e di droni il FirstAverageTime ottiene un tempo di visita medio leggermente più basso di quello del FirstMaxTime ma un tempo massimo più alto in quanto tende a non riassegnare dei blocchi (magari quelli più ai margini) che hanno pochi punti con un tempo di visita molto alto ma molti con tempo di visita molto basso. Al contrario il FirstMaxTime al costo di aumentare leggermente il tempo medio di visita, riesce a diminuire sensibilmente il tempo di visita massimo dimostrandosi più equo.