1. Documentazione software SciaDro ver. 2.0

1.1. Analisi

Si sono identificati i seguenti elementi fondamentali del simulatore SciaDro:

- *Ambiente* – area di esplorazione su cui sono posizionati gli altri elementi;
- *Drone* – entità in movimento che porta a termine l'obiettivo prefissato dalla missione;
- *Target* – entità che deve essere rilevata dai droni;
- *Ostacolo* – entità che deve essere evitata dai droni;
- *Stigmergia* – meccanismo di comunicazione indiretta usato dai droni;
- *Flocking* – meccanismo di coordinamento collettivo usato dai droni.

5.1.2. Ambiente

L'ambiente può essere semplicemente considerato come un'area bidimensionale vincolata dove il drone può muoversi in modo continuo. L'ambiente risulta discretizzato sia da un punto di vista spaziale che temporale e può contenere sia target, con posizione statica o dinamica, che ostacoli, con posizione statica.

5.1.3. Drone

I droni, o UAVs (Unmanned Aerial Vehicles), rappresentano le uniche entità in grado di muoversi in modo continuo all'interno dell'area di esplorazione. In sostanza i droni, organizzandosi in flocks, ovvero in gruppi coesi più o meno numerosi che viaggiano nella stessa direzione, hanno l'obiettivo di trovare i target presenti nell'area di esplorazione entro il tempo di autonomia della batteria, evitando sia gli ostacoli che gli altri droni e usando la stigmergia come meccanismo di auto-coordinamento durante la ricerca.

5.1.4. Target

I target si trovano in posizioni sconosciute a priori e devono essere scoperti dai droni entro il loro tempo di autonomia. Un target può essere considerato come "trovato" (found) quando un drone, equipaggiato con una tecnologia di sensing adeguata, si trova in posizione tale da poterlo rilevare.

Esempi di target sono una mina antiuomo inesplosa, una discarica abusiva, un principio d'incendio, e così via.

5.1.5. Ostacolo

Un ostacolo è definito come un'entità statica all'interno dell'ambiente di simulazione, che non può essere attraversata dai droni. Costruzioni e alberi rappresentano tipici esempi di ostacoli che possono trovarsi all'interno di un contesto applicativo. Il numero, la forma e la dimensione degli ostacoli dipendono dalla quota di volo dei droni: tipicamente, maggiore è la quota di volo, minore sarà il numero di ostacoli.

5.1.5.1. Modulo di collision avoidance

Tipicamente, l'ambiente di esplorazione prevede la presenza di ostacoli e il numero di droni dello sciame potrebbe essere elevato. Di conseguenza, risulta strettamente necessario un meccanismo di collision avoidance al fine di rendere più realistica la simulazione. Il modulo di collision avoidance può essere suddiviso in due sottomoduli:

1) Modulo di obstacle avoidance;
2) Modulo di overlap avoidance.

5.1.5.1.1. Modulo di obstacle avoidance

L'obiettivo principale di questo modulo software è quello di evitare le collisioni tra i droni e gli ostacoli presenti nell'ambiente di esplorazione. In un contesto applicativo reale, questo meccanismo risulta strettamente necessario per ragioni di sicurezza e di costi.

5.1.5.1.2. Modulo di overlap avoidance

L'obiettivo di questo modulo è quello di evitare che si verifichino collisioni (o scontri) tra due o più droni in movimento. Questa condizione deve essere sempre verificata, in particolare quando il numero di droni dello sciame aumenta.

5.1.6. Stigmergia

La stigmergia è un meccanismo bio-ispirato usato da formiche, api ed altri insetti sociali, ad esempio durante l'attività di foraggiamento. La stigmergia è un meccanismo emergente, perché entità con limitate capacità computazionali possono organizzarsi in maniera distribuita e robusta.

L'elemento fondamentale della stigmergia è il feromone, ovvero una sostanza chimica in grado di attrarre altre entità nei luoghi in cui viene rilasciata. Il feromone ha una durata limitata nel tempo e dopo un certo periodo scompare a causa dell'evaporazione. Inoltre, un'entità attratta dal feromone è soggetta all'assuefazione olfattiva: dopo un certo tempo l'attrazione esercitata dal feromone perde il suo effetto.

Il meccanismo di stigmergia, così come descritto, è stato progettato, implementato e testato nel simulatore. Lo scopo principale è quello di attrarre altre entità su un target di interesse comune, perché, in questo modo, altri target vicini possono essere facilmente scoperti.

Il feromone può essere anche repulsivo: invece di attrarre entità in un'area delimitata, respinge le entità vicine.

5.1.7. Flocking

Il flocking è il comportamento esibito quando un gruppo di uccelli, denominato flock, sta facendo attività di foraggiamento oppure è in volo. Il flocking è simile al comportamento di shoaling dei pesci, al comportamento di swarming degli insetti e ai comportamenti di altri animali sociali. La ragione principale per organizzare le entità in flocks sta nel fatto di raggrupparle per rendere più efficiente ed affidabile il processo di scoperta dei target. Infatti, un flock costituito da più droni ha un raggio di sensing più ampio rispetto al singolo drone. Il meccanismo di flocking può essere ulteriormente suddiviso in tre diversi sotto-meccanismi:

1) Separazione;
2) Allineamento;
3) Coesione.

5.1.7.1. Separazione

Il meccanismo di separazione ha l'obiettivo di distanziare i droni. I compiti principali del meccanismo di separazione sono i seguenti:

- Prevenire le collisioni tra i droni;
- Avere dei flock più ampi, quindi avere una maggiore copertura di area di sensing.

5.1.7.2. Allineamento

Il meccanismo di allineamento ha lo scopo di allineare il flock in una direzione. Fondamentalmente, esso consiste nel ruotare verso la direzione media del flock al fine di seguire in modo più accurato la direzione dei compagni di flock.

5.1.7.3. Coesione

Questo meccanismo ha lo scopo di recuperare il flock e si attiva quando un'entità è "troppo lontana" dal resto del flock. La lontananza dal resto del flock è un parametro regolabile.

5.1.8. Valutazione della performance del simulatore SciaDro

Lo scopo del simulatore SciaDro non è solo quello di simulare il coordinamento di sciami di droni attraverso flocking e stigmergia, ma anche quello di valutare la performance al variare dei parametri in ingresso.

I parametri in ingresso e in uscita sono stati definiti nella fase di progettazione, dove ogni componente del software è stato modellato e progettato. Inoltre, i meccanismi di stigmergia e flocking possono essere attivati e disattivati singolarmente, al fine di valutare la performance con (o senza) uno o entrambi i meccanismi.

Lo scopo principale della valutazione delle performance del simulatore SciaDro è quello di quantificare l'impatto della stigmergia o del flocking o di entrambi i meccanismi sul tempo medio di rilevamento del 95% dei target.

1.2. Progettazione

Nella fase di progettazione, si sono fatte alcune assunzioni sull'ambiente, sul drone e sulle altre entità coinvolte nella simulazione. In particolare, queste assunzioni riguardano le caratteristiche principali dell'ambiente e delle entità stesse.

5.2.1. Progettazione dell'ambiente

L'ambiente preso in considerazione non è quello reale, ma è un ambiente bidimensionale strutturato e limitato, in cui lo spazio e il tempo risultano entrambi discretizzati.
Lo spazio è simulato da una griglia di $N \times M$ celle quadrate di lato S. Il tempo è discretizzato in intervalli temporali: $t_0 + T$, $t_0 + 2T$, …, $t_0 + NT$. T è definito come *tick*.

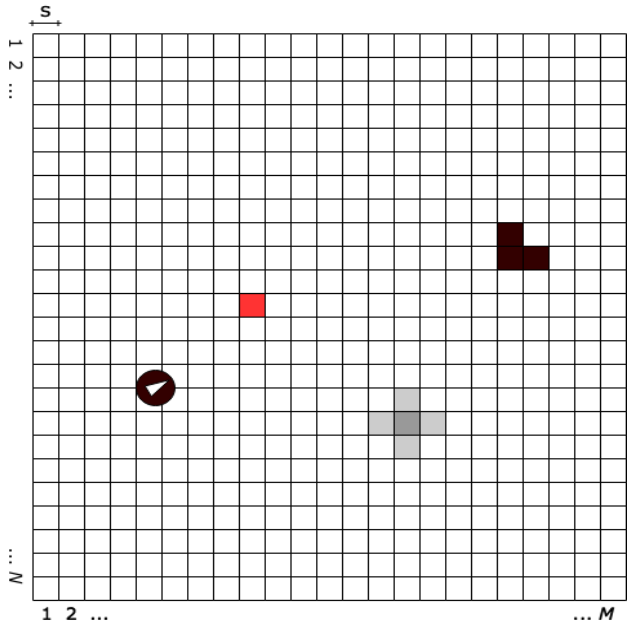

Figura 1 - Definizione dell'ambiente simulato

L'ambiente simulato contiene i seguenti elementi:

- Droni;
- Target;
- Ostacoli;
- Feromoni digitali.

In questa versione del simulatore, si è assunto per semplicità:

S = 1 metro

T = 1 secondo

Naturalmente, la lunghezza del lato di una patch può non essere sempre uguale ad un metro. In questo caso è necessaria una conversione da metri a numero di patches. Lo stesso discorso vale per la conversione da secondi a ticks.

Supponendo di avere una patch di lato S diverso da un metro, può essere sfruttata la seguente formula di conversione per determinare il numero di patches a cui corrisponde un metro: 1 *m* = 1/S *patches*

Allo stesso modo, si può determinare a quanti ticks corrisponde un secondo: 1 *s* = 1/T *ticks*

Seguendo questa linea di ragionamento, si può determinare la formula di conversione da metri/secondo a patches/tick:

1 m/s = (1/S)(1/T) patches/tick

Ad ogni tick, il drone effettua rilevamenti di ostacoli, targets e feromoni vicini alla sua posizione ed esegue regole comportamentali. Lo sciame di droni si muove separatamente, organizzato in diversi flocks.

Il blocco fondamentale dello spazio discretizzato simulato è rappresentato dalla cella. Ogni cella (denominata anche patch) ha delle coordinate xy nello spazio bidimensionale. Questa può contenere o un target o un ostacolo. Inoltre, una cella può essere prenotata da un drone per via del meccanismo di collision avoidance.

Un drone può rilasciare su una patch un feromone digitale. La variabile "delta evaporation" deve essere memorizzata nella patch contenente feromone digitale, al fine di conoscere la quantità di feromone stesso che deve essere sottratta ad ogni tick.

Considerando l'elemento patch come una classe, la Figura 34 illustra la struttura di un oggetto patch.

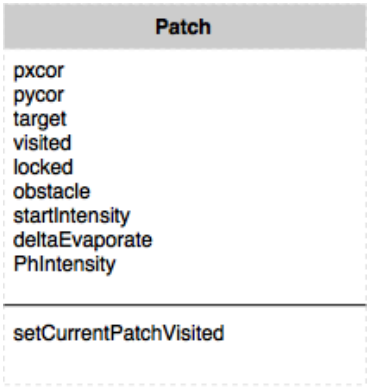

Figura 2 - Struttura della classe Patch

5.2.2. Progettazione del Drone

Il drone è l'elemento in grado di muoversi in modo continuo all'interno dello spazio bidimensionale di ricerca. L'obiettivo principale del drone è quello di scoprire nuovi target. Inoltre, il drone deve evitare di scontrarsi con gli ostacoli e con gli altri droni e, insieme a loro, deve organizzarsi in flock e completare il processo di scoperta dei target entro il tempo di autonomia, usando la stigmergia come meccanismo di comunicazione indiretta.

Gli scontri e le collisioni con gli altri droni e con gli ostacoli sono, naturalmente, simulati. Per questo motivo, a partire dalla fase di progettazione, questi eventi possono essere denominati "sovrapposizioni tra droni e droni" e "sovrapposizioni tra droni e ostacoli", rispettivamente. Si deve notare che il sensing potrebbe essere influenzato dalla velocità del drone. Esiste una velocità massima per cui la performance della tecnologia di sensing non risulta influenzata dalla velocità del drone: tale velocità è denominata *velocità di crociera* e risulta minore della velocità massima.

Lo stato del drone è caratterizzato da:

- Posizione nell'ambiente (drone.x e drone.y);
- Numero di droni vicini nello sciame (drone.flockmates);
- Velocità di crociera (drone.cruisingSpeed);
- Velocità corrente (drone.speed);
- Direzione corrente (drone.heading);
- Velocità angolare corrente (drone.angularSpeed);
- Autonomia (drone.endurance).

Il drone esegue azioni al fine di:

1) Evitare ostacoli
    a) Collision avoidance
2) Coordinarsi con gli altri droni attraverso il meccanismo di flocking
    a) Separate
    b) Align
    c) Cohere
3) Rilasciare il feromone
4) Muoversi nell'ambiente per ricercare i targets
    a) Movement
    b) Accelerate
    c) Decelerate

5.2.2.1. Modellazione del drone

Il drone può essere modellato come un cerchio in movimento nello spazio bidimensionale. I parametri associati al drone dipendono dalle sue caratteristiche fisiche e architetturali.

Al fine di rilevare il target, è stato necessario inserire un codice di sensing. L'area del cono di sensing dipende prevalentemente dal tipo di sensore.

Per le ragioni sopra menzionate, tutti questi parametri devono essere fissati prima di avviare la simulazione.

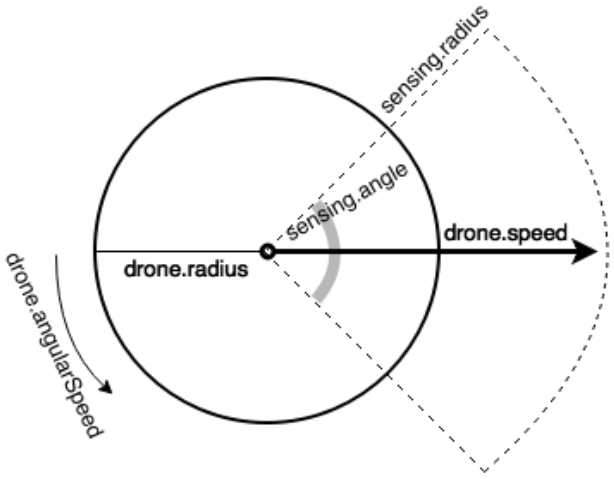

Figura 3 - Modello parametrico del drone

La seguente tabella riassume i parametri strutturali che caratterizzano il drone.

Parametro	**Definizione**	**Unità di misura**
drone.radius	Raggio fisico	(m)
sensing.radius	Raggio di sensing	(m)
sensing.angle	Angolo di sensing	(gradi)
drone.cruisingSpeed	Velocità di crociera	(m/s)
drone.speedMax	Velocità massima	(m/s)
drone.speed	Velocità corrente	(m/s)
drone.acceleration	Accelerazione	(m/s^2)
drone.deceleration	Decelerazione	(m/s^2)

drone.endurance	Autonomia	(minuti)
drone.heading	Direzione corrente	(gradi)
drone.accelerationAng	Accelerazione angolare	(rad/s^2)
drone.decelerationAng	Decelerazione angolare	(rad/s^2)
drone.velocityAngularMax	Velocità angolare massima	(rad/s)
drone.angularSpeed	Velocità angolare corrente	(rad/s)

Tabella 1 - Parametri strutturali del drone

5.2.2.2. Diagramma delle classi del drone

Ad ogni ciclo di aggiornamento, il drone è posizionato in un punto specifico nello spazio bidimensionale di ricerca. Esso può essere equipaggiato con uno o più sensori e può muoversi in modo longitudinale e rotazionale. Inoltre, è anche in grado di rilevare i target e di rilasciare impronte di feromone.

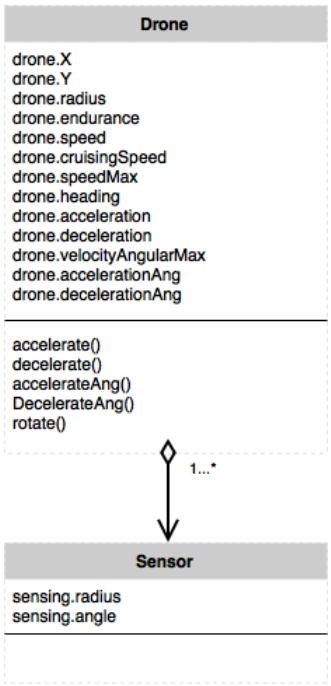

Figura 4 - Drone e sensori

5.2.3. Progettazione del flocking

Di seguito si illustra la progettazione del meccanismo di flocking descritto nella fase di analisi. Il drone distingue tre diverse aree di prossimità: l'area di separazione (separate), l'area di allineamento (align) e l'area di coesione (cohere). Ciascuna di tali aree ha uno specifico raggio ed è associata ad uno specifico comportamento del drone. Le tre aree di prossimità hanno in comune l'angolo *drone.flocking.angle*.

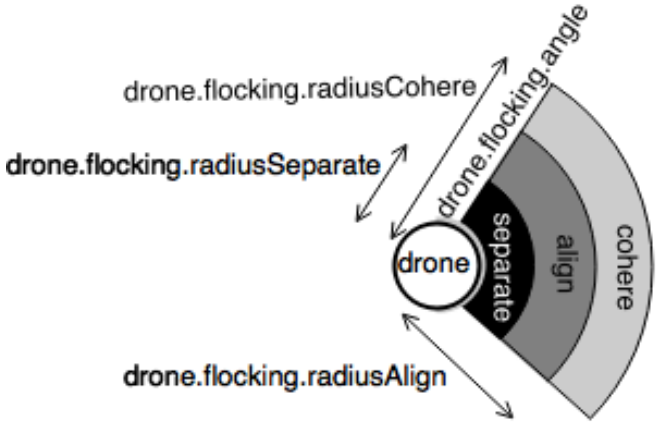

Figura 5 - Aree del flocking

I parametri illustrati nella figura 37, sono dettagliati nella tabella seguente.

Parametro	**Definizione**	**Unità di misura**
drone.flocking.angle	Angolo di visione del flock	(gradi)
drone.flocking.radiusCohere	Raggio di coesione $\in$ (*drone.flocking.radiusAlign*, $+inf$)	(m)
drone.flocking.maxCohereTurn	Massimo angolo di rotazione nel mecanismo di coesione	(gradi)
drone.flocking.radiusAlign	Raggio di allineamento $\in$ (*drone.flocking.radiusSeparate*, *drone.flocking.radiusCohere*)	(m)

drone.flocking.maxAlignTurn	Angolo di rotazione massimo nel mecanismo di allineamento	(gradi)
drone.flocking.radiusSeparate	Raggio di separazione $\in$ (*drone.collisionVision*, *+inf*)	(m)
drone.flocking.maxSeparateTurn	Angolo di rotazione massimo nel meccanismo di separazione	(gradi)
drone.flocking.wiggleVar	Numero casuale $\in$ (0, N)	adimensionale

Tabella 2 - Parametri strutturali del flocking

5.2.3.1. Comportamento separate

Se uno o più droni si trovano all'interno dell'area identificata dal raggio *drone.flocking.radiusSeparate*, allora il drone realizza una rotazione per separarsi dagli altri droni. Tale rotazione è limitata dall'angolo massimo di rotazione *drone.flocking.maxSeparateTurn*.

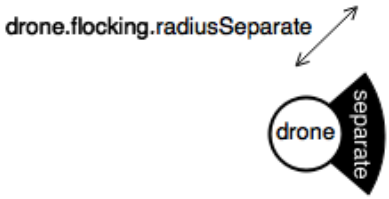

Figura 6 - area di riferimento per il comportamento separate

5.2.3.2. Comportamento align

Il drone verifica la presenza di altri droni nell'area di align e ruota di un angolo *drone.flocking.alignAngle* più un angolo casuale *drone.flocking.wiggleVar* al fine di allinearsi al flock. La rotazione di allineamento è limitata da *drone.flocking.maxAlignTurn*.

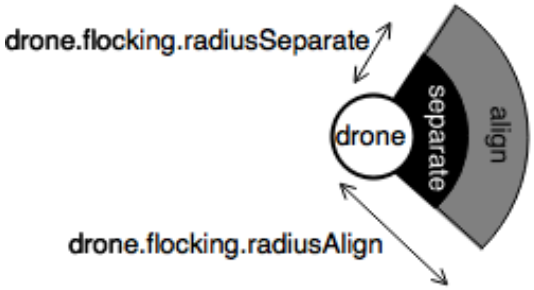

Figura 7 - area di riferimento per il comportamento align

5.2.3.3. Comportamento cohere

Il drone calcola il baricentro dei compagni di flock presenti nell'area di cohere, il cui raggio è *drone.flocking.radiusCohere*, e ruota in quella direzione. La massima rotazione permessa è un angolo pari a *drone.flocking.MaxCohereTurn* più una quantità random *drone.flocking.wiggleVar*.

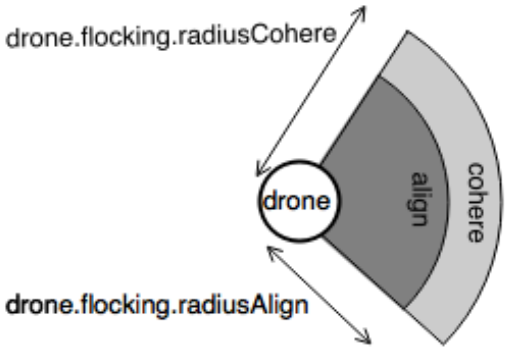

Figura 8 - area di riferimento per il comportamento cohere

5.2.3.4. Algoritmo di flocking

Come già illustrato graficamente, il meccanismo di flocking è costituito da tre sottomeccanismi a priorità decrescente:

- Separate;
- Align;
- Cohere.

Ad ogni ciclo di aggiornamento (tick) può essere eseguito solo uno di questi tre meccanismi. La figura 41 rappresenta il diagramma di attività dell'algoritmo di flocking, dove la variabile casuale wiggle non è stata inserita per semplicità.

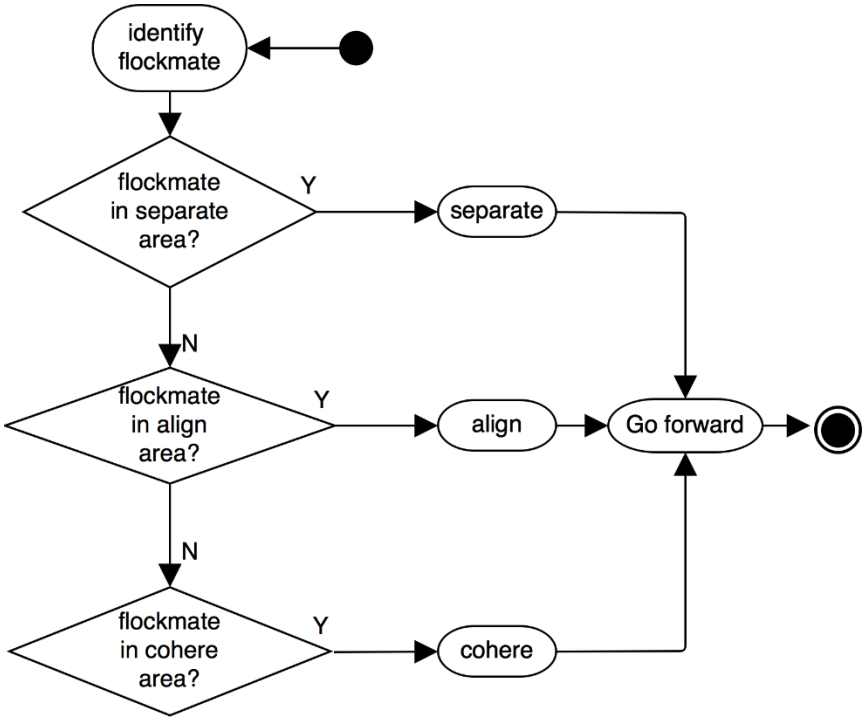

Figura 9 - Diagramma di attività dell'algoritmo di flocking

5.2.3.5. Diagramma di classe flocking

Tutti i parametri di flocking sono fissati prima di avviare la simulazione ed influenzano il movimento del drone e dunque il comportamento di flock. Di conseguenza, l'intera simulazione è influenzata dal comportamento di flocking.

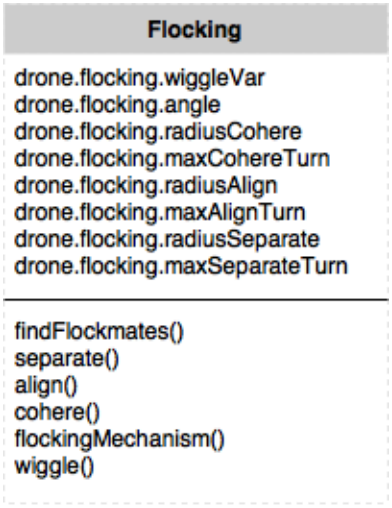

Figura 10 – Diagramma della classe Flocking

5.2.4. Progettazione del meccanismo di collision avoidance

Il meccanismo di collision avoidance risulta obbligatorio perché i droni non devono scontrarsi con gli ostacoli o tra di loro. Il meccanismo di collision avoidance può essere suddiviso in due sotto meccanismi:

1) Obstacle avoidance
2) Overlapping avoidance

Il primo meccanismo permette di evitare le collisioni tra un drone ed un ostacolo. Il secondo permette di evitare le collisioni tra due droni.

5.2.4.1. Modello di obstacle avoidance

Uno degli obiettivi principali del drone è quello di evitare gli ostacoli che incontra sulla sua traiettoria di volo. A tal fine, si è progettato, implementato e testato un modello di obstacle avoidance.

Il modello segue una semplice regola: trovare il gap tra ostacoli che comporta la minima rotazione del drone verso il gap stesso. La semplicità di questa regola è dovuta al fatto che il drone ha un tempo limitato per riconoscere l'ostacolo, trovare il gap e ruotare verso la sua direzione.

In un contesto reale, il drone può trasportare un array di sensori ad infrarossi in grado di riconoscere le distanze e gli angoli degli ostacoli vicini. L'attività di riconoscimento di un ostacolo è modellata così come fatto per il sensing: un cono con un certo angolo e un certo raggio. Il drone, dopo aver calcolato le distanze e gli angoli degli ostacoli vicini, ricerca il gap che comporta la minore rotazione rispetto alla direzione corrente.

A causa della grande varietà di ostacoli che si possono trovare nei diversi contesti applicativi, sia la dimensione del gap che il raggio e l'angolo di visione degli ostacoli risultano configurabili. Durante la simulazione, questi parametri influenzano il comportamento del drone, perché l'algoritmo di obstacle avoidance dipende dal loro valore.

Parametro	**Definizione**	**Unità di misura**
drone.collision.gapAngle	Minimo angolo che permette al drone di passare attraverso il varco tra due ostacoli ∈ (0, *drone.sight.angleMax*)	(gradi)
drone.collision.vision	Raggio di visibilità degli ostacoli	(m)

drone.sight.angleMax	Angolo di visibilità massimo degli ostacoli	(gradi)

Tabella 3 - Parametri strutturali del meccanismo di obstacle avoidance

5.2.4.2. Algoritmo di obstacle avoidance

Il drone, durante la simulazione, esplora l'ambiente di ricerca al fine di rilevare i target. Per mantenere la sua operatività e integrità, il drone deve evitare gli ostacoli. La figura 43 fornisce un'idea del funzionamento del meccansimo di obstacle avoidance.

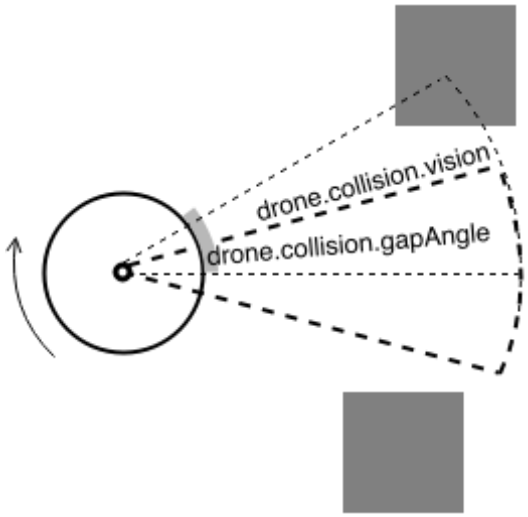

Figura 11 - Principio di funzionamento del meccanismo di collision avoidance

L'algoritmo opera nel seguente modo:

- Prima di controllare la presenza di ostacoli, il drone deve verificare se è fermo sulla stessa posizione da un certo periodo di tempo, o, in termini di simulazione, da un certo numero di ticks consecutivi.
- Se risulta bloccato, significa che l'ostacolo presente di fronte al drone è probabilmente un muro che gli impedisce di avanzare. Al fine di evitare lo stallo, il drone esegue un movimento rotazionale di 90° più una certa quantità casuale *drone.flocking.wiggleVar*.
- Viceversa, se il drone non è bloccato, allora controlla la presenza di ostacoli.
- Se non ci sono ostacoli, significa che il drone può avanzare senza alcun problema.
- Se sono rilevati degli ostacoli, il drone controlla la presenza del varco tra essi, in riferimento ad un certo angolo di gap.
- Se il varco non viene trovato, allora il drone ruota leggermente o a destra o a sinistra (la direzione viene scelta casualmente), al fine di ricercare un altro gap vicino. Questo angolo di rotazione è pari alla metà di *drone.sight.angleMax*. Successivamente, il drone effettua una decelerazione, perché si trova di fronte a ostacoli senza varco.

- Se il varco viene trovato, allora il drone esegue una rotazione verso il gap. Anche in questo caso il drone decelera, perché il varco potrebbe essere stretto e dunque il drone deve attraversarlo con attenzione.

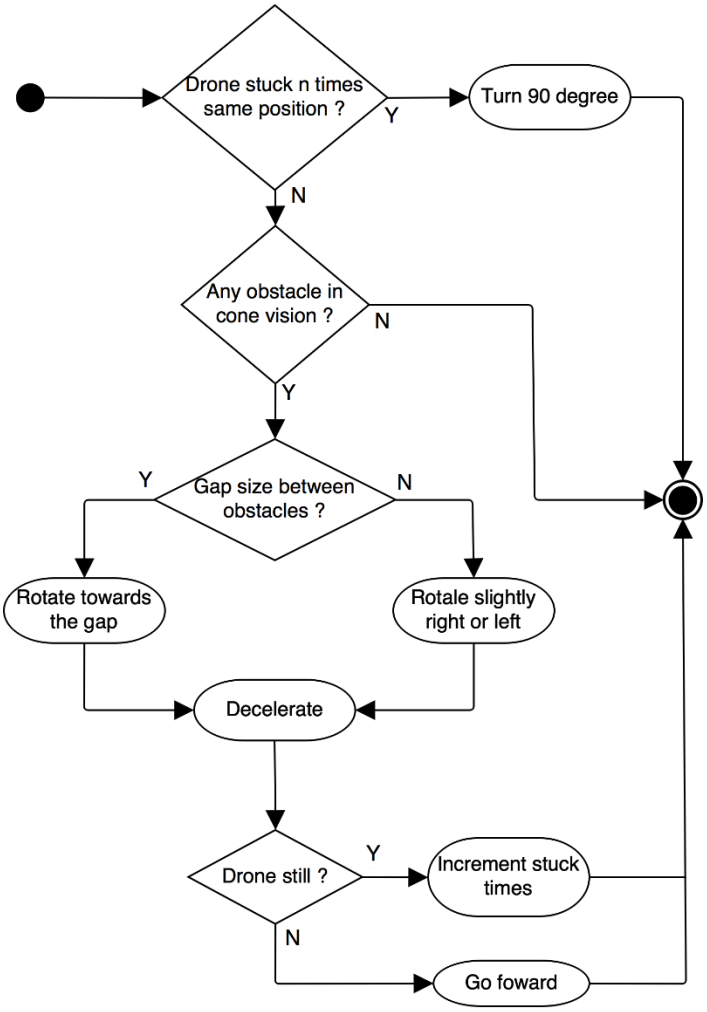

Figura 12 - Diagramma di attività del meccanismo di obstacle avoidance

5.2.4.3. Diagramma di classe obstacle avoidance

La figura 45 illustra il diagramma di classe relativo al meccanismo di obstacle avoidance.

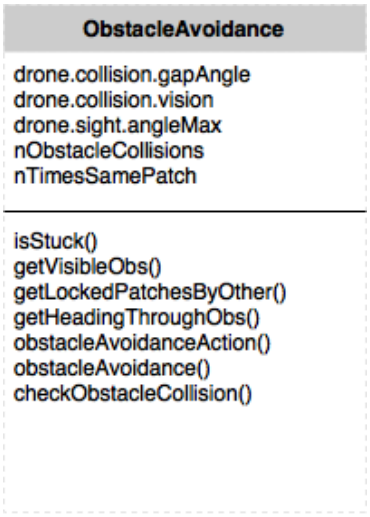

Figura 13 - Diagramma di classe ObstacleAvoidance

I primi tre attributi presenti nel diagramma sono configurati prima di avviare la simulazione. Gli ultimi due sono usati, rispettivamente, per contare il numero di collisioni con gli ostacoli e il numero di ticks durante il quale il drone rimane bloccato.

5.2.4.4. Modello di overlapping avoidance

Il solo meccanismo di obstacle avoidance non è sufficiente per evitare le sovrapposizioni tra i droni, perché un drone non può prevedere i movimenti degli altri droni. Al fine di evitare questo tipo di problema, si è progettato un modello di overlapping avoidance tra droni: conoscendo l'accelerazione, la decelerazione, la velocità massima e la rotazione massima di un drone nell'arco di una finestra temporale, si può calcolare la regione di raggiungibilità (reachable region), ovvero la regione che, in termini di patches, il drone potrebbe ricoprire nel prossimo futuro, o da un punto di vista della simulazione, nei prossimi ticks.

Come già specificato, l'ambiente risulta discretizzato sia da un punto di vista spaziale che temporale. Fondamentalmente, la strategia per evitare le sovrapposizioni tra i droni è la seguente: in primo luogo, schedulare una data regione di raggiungibilità, precisamente un cono, e, in secondo luogo, trattare le regioni già schedulate dagli altri droni come ostacoli, applicando il meccanismo di obstacle avoidance. In questo modo, le potenziali regioni in cui un drone può trovarsi nell'immediato futuro non sono accessibili dagli altri droni.

Anche in questo caso, il meccanismo di overlapping avoidance dei droni risulta configurabile: il numero di droni può variare notevolmente tra i diversi scenari applicativi e le caratteristiche tecniche come velocità massima, massima rotazione e così via, risultano molto diverse tra modelli di drone differenti. La posizione schedulata da un drone per evitare le collisioni è semplicemente un cono con un certo

raggio e un certo angolo. I due parametri configurabili che determinano il cono da schedulare sono riportati nella seguente tabella.

Parametro	Definizione	Unità di misura
drone.reachable.angle	Angolo delle patches schedulate	(gradi)
drone.reachable.radius	Raggio delle patches schedulate	(m)

Tabella 4 - Parametri strutturali del meccanismo di overlapping avoidance

5.2.4.5. Algoritmo di overlapping avoidance

L'algoritmo di overlapping avoidance tra i droni è suddiviso in due fasi descritte di seguito:

1. Lo scheduling delle patches;
2. Il meccanismo per evitare le sovrapposizioni.

Lo scheduling delle patches prevede le casistiche enumerate di seguito:

a) Nel caso generale, il drone vede le patches libere all'interno del cono di raggiungibilità e quelle sotto la sua posizione e le blocca;

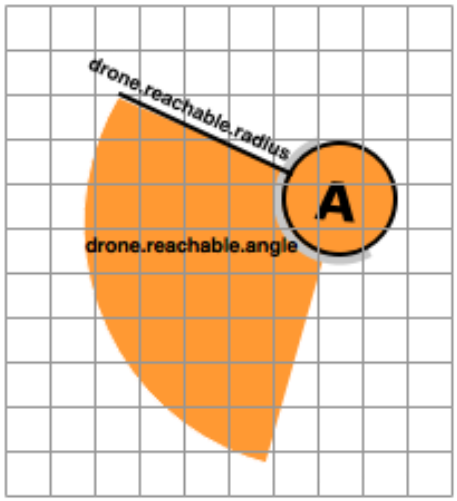

Figura 14 - Caso generale: selezione del cono di raggiungibilità

b) In altri casi il cono di scheduling può essere sovrapposto. Nella figura 47 il drone A blocca le patches nel cono arancione. Di conseguenza, il drone B vede le patches bloccate dal drone A e

blocca esclusivamente quelle patches del suo cono di scheduling che non occupate, ovvero le patches nel cono blu.

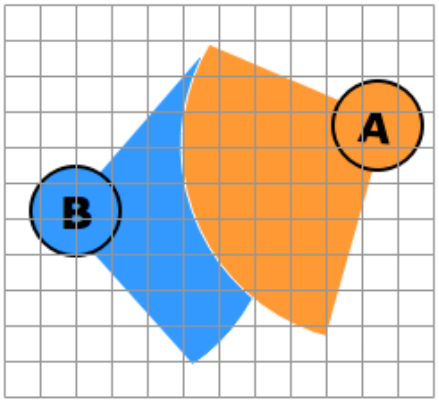

Figura 15 - Caso di sovrapposizione dello scheduling delle patches

c) I casi a) e b) potrebbero portare ad un problema di deadlock. Il deadlock può verificarsi se un drone A che si trova dietro ad un drone B, schedula delle patches davanti al drone B. In questo caso il drone B non può avanzare perché vede la posizione già schedulata: il drone B risulta bloccato da un ostacolo virtuale creato dal drone A.

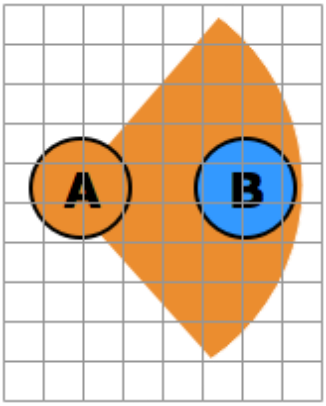

Figura 16 - Problema di deadlock

Al fine di evitare questa situazione, risulta necessario aggiungere un'ulteriore requisito: un drone, prima di schedulare le possibili posizioni future, deve controllare se all'interno del suo cono di scheduling è presente o meno un altro drone. Nel caso in cui ci sia, allora blocca solo le patches al di sotto della sua posizione.

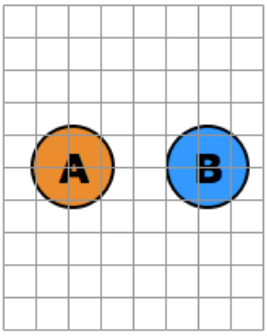

Figura 17 - Soluzione al problema di deadlock

La figura 50 chiarisce l'algoritmo di scheduling della regione di raggiungibilità attraverso un semplice diagramma di attività.

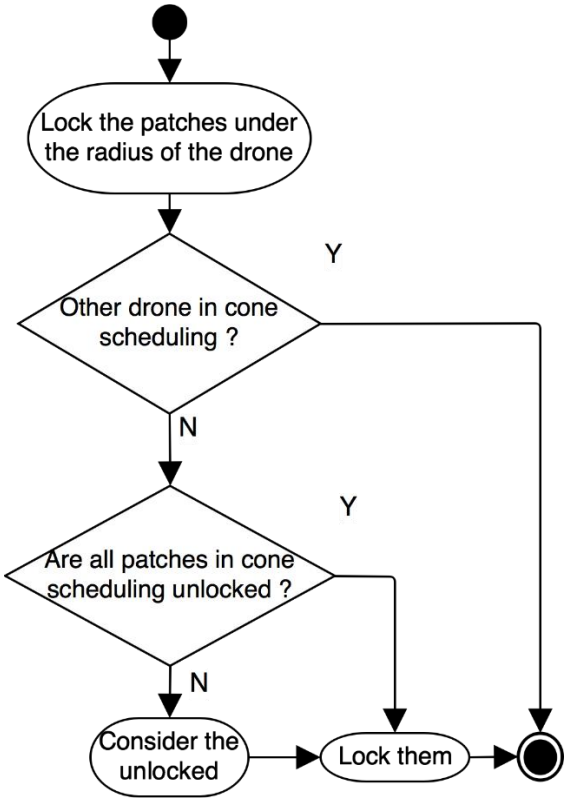

Figura 18 - Diagramma di attività dello scheduling della regione di raggiungibilità

5.2.4.6. Diagramma di classe overlapping avoidance tra droni

Gli attributi relativi al raggio e all'angolo delle patches bloccate devono essere configurati prima di avviare la simulazione e dipendono dalla velocità di crociera assunta dal drone. I rimanenti parametri cambiano durante la simulazione.

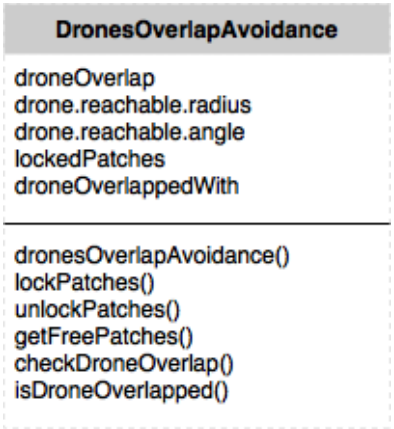

Figura 19 - Diagramma di classe DronesOverlapAvoidance

5.2.4. Progettazione dell'impronta di feromone attrattivo

Un drone rilascia un'impronta di feromone attrattivo quando rileva un target, al fine di "richiamare" altri droni attivando il meccanismo di coordinamento basato su stigmergia.

Un'impronta di feromone è modellata come un tronco di cono e tipicamente ricopre più celle dello spazio di ricerca. Più impronte di feromone rilasciate in prossimità spazio-temporale si aggregano a formare una traccia feromonica. La traccia feromonica evapora nel tempo.

L'impronta feromonica è caratterizzata da un raggio centrale, da un raggio periferico (*mark.RadiusTop* e *mark.radiusDown* rispettivamente) e da un tasso di diffusione (*track.diffRate*). La traccia feromonica è caratterizzata da un tasso di evaporazione superficiale (*track.evapRate*).

Le funzioni eseguite sulla traccia feromonica sono:

- Evaporazione (Evaporate);
- Diffusione (Opzionale).

5.2.4.1. Impronta e traccia feromonica

Di seguito viene illustrato il modello dell'impronta e della traccia di feromone attrattivo, insieme ai parametri caratteristici.

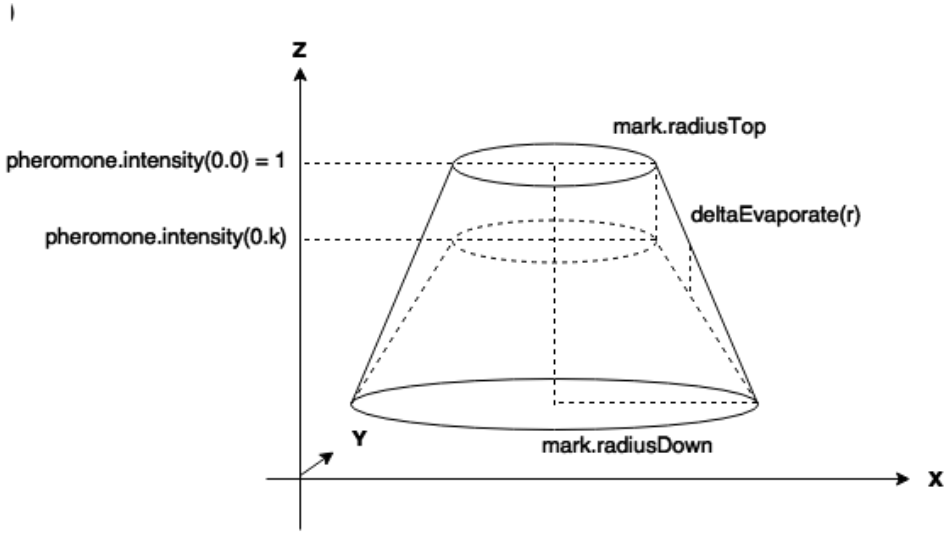

Figura 20 - Modello tridimensionale dell'impronta di feromone attrattivo

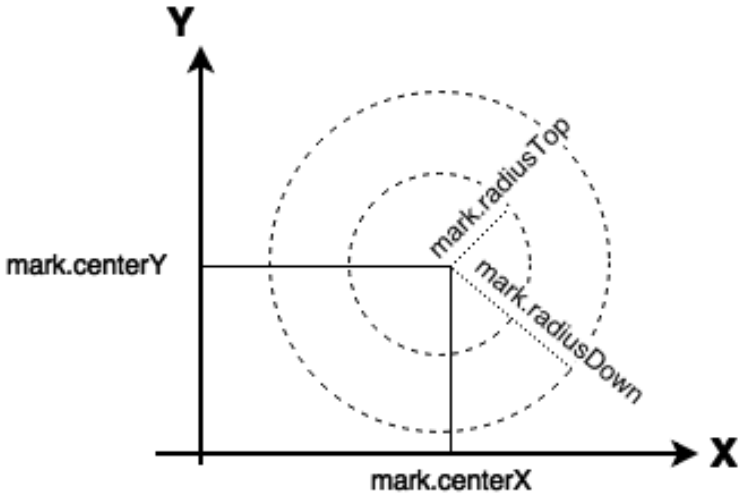

Figura 21 - Modello bidimensionale dell'impronta di feromone attrattivo

Parametro	Definizione	Unità di misura
mark.centerX	Coordinata X del centro di rilascio dell'impronta di feromone attrattivo	-
mark.centerY	Coordinata Y del centro di rilascio dell'impronta di feromone attrattivo	-
mark.radiusTop	Raggio di rilascio iniziale dell'impronta di feromone ∈ [0, +*inf*)	(m)
mark.radiusDown	Raggio di rilascio diffuso iniziale dell'impronta di feromone ∈ [*mark.radiusTop*, +*inf*)	(m)
track.evapRate	Tasso di evaporazione della traccia di feromone ∈ [0, 1]	adimensionale

Tabella 5 - Parametri strutturali dell'impronta di feromone

L'intensità di rilascio dell'impronta di feromone attrattivo al tick 0 è pari a:

$$pheromone.intensity\ (r, 0) = 1$$

con $r < mark.radiusTop$

L'intensità di feromone decresce a partire da *mark.radiusTop* fino a *mark.radiusDown* in accordo alla seguente espressione:

$$pheromone.intensity\ (r,k) = \ pheromone.intensity\ (0,k) * \frac{(r - mark.radiusDown)}{(mark.radiusTop - mark.radiusDown)}$$

con $mark.radiusTop < r < mark.radiusDown$, $k \geq 0$

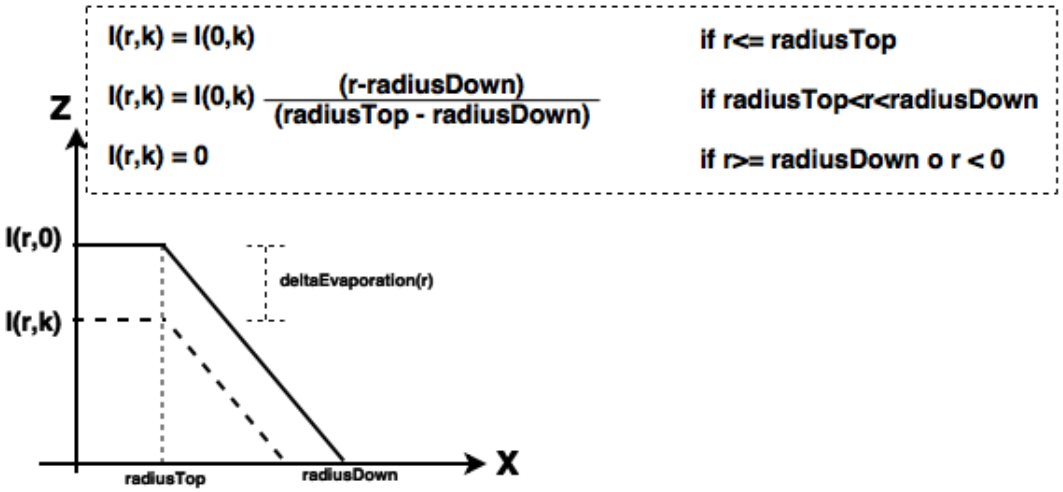

Figura 22 - Modello dell'intensità di feromone e comportamento di evaporazione

5.2.4.2. Comportamento di evaporazione

L'evaporazione rappresenta il decadimento lineare della traccia di feromone nel tempo. Ad ogni ciclo di aggiornamento il feromone evapora di una certa quantità a partire da quando viene rilasciato. Dopo un certo tempo, a seconda del valore del tasso di evaporazione, il feromone scompare completamente.

Dal momento che il tempo simulato è discretizzato e l'evaporazione è lineare, ad ogni ciclo di aggiornamento viene sottratta da ogni patch sempre la stessa quantità di feromone. Ad esempio, se il tasso di evaporazione è pari a 0.1, dopo 10 ticks, il feromone scompare completamente indipendentemente dalla sua intensità iniziale.

L'espressione matematica che descrive l'evaporazione feromonica è la seguente:

$$pheromone.intensity\ (r,k) = \ pheromone.intensity\ (r,k-1) - \ deltaEvaporation\ (r)$$

dove

- $deltaEvaporation\ (r) = \ evapRate * pheromone.intensity\ (r,0)$, ovvero la quantità di feromone che evapora ad ogni tick;
- $pheromone.intensity\ (r,k)$, l'intensità dell'impronta feromonica in funzione della distanza r dal centro del rilascio e del tick k;

- $pheromone.intensity\ (r, 0)$, l'intensità dell'impronta feromonica al momento del rilascio in funzione della distanza r dal centro del rilascio.

5.2.4.3. Meccanismo di coordinamento basato su stigmergia

Come già evidenziato in precedenza, la stigmergia è un meccanismo emergente. Tale meccanismo bioispirato è stato progettato e, nella fase successiva, implementato.

Esso segue due semplici regole:

- se un drone rileva un target, vi rilascia sopra un'impronta feromonica;
- se un drone rileva un'impronta feromonica, segue la direzione corrispondente al picco di intensità.

Il meccanismo di coordinamento basato su stigmergia ha la potenzialità di attrarre gruppi di droni che sono posizionati in prossimità spazio-temporale rispetto alla traccia feromonica.

5.2.4.4. Assuefazione olfattiva

Un altro importante meccanismo da evidenziare è rappresentato dall'assuefazione olfattiva (*olfactory habituation*). Un'agente, dopo aver annusato del feromone per un certo periodo di tempo, ne diventa assuefatto, ovvero non è più in grado di percepirne la presenza. Di conseguenza il drone, dopo aver annusato il feromone attrattivo per un certo numero di ticks, non ne viene più influenzato.

L'assuefazione olfattiva risulta importante per due ordini di ragioni:

- evita di attrarre un numero eccessivo di agenti sullo stesso target;
- evita di rimanere nei pressi dello stesso target per troppo tempo.

Questi due aspetti permettono di ridurre la probabilità di collisioni ed il tempo impiegato nella ricerca di target in prossimità spaziale.

L'assuefazione olfattiva è caratterizzata da un parametro che esprime la quantità di tempo entro la quale un drone deve annusare il feromone prima di ignorarlo. Questo è un parametro configurabile in modo da poter essere modificato in funzione dello specifico contesto applicativo. Il seguente diagramma di attività illustra il meccanismo di funzionamento dell'assuefazione olfattiva.

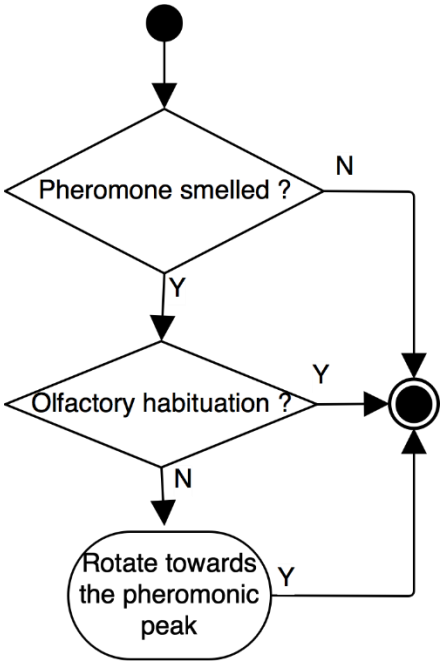

Figura 23 - Diagramma di attività relativo al funzionamento dell'assuefazione olfattiva

5.2.4.5. Diagramma di classe stigmergia attrattiva

Nel seguente diagramma di classe sono evidenziati tutti gli attributi e i metodi relativi al feromone attrattivo.

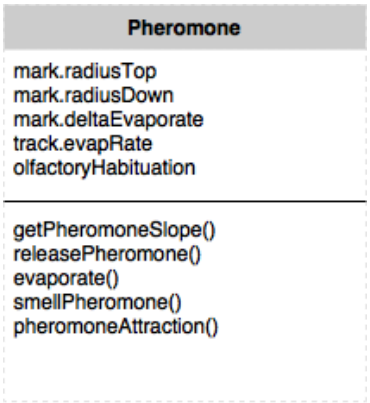

Figura 24 - Diagramma di classe stigmergia attrattiva

Gli attributi radiusTop, radiusDown, evapRate ed olfactoryHabituation devono essere configurati prima dell'inizio della simulazione.

5.2.5. Elemento ostacolo

Un ostacolo è un elemento che ostruisce il passaggio di un drone. Esso ha una posizione statica e occupa una cella (o patch) all'interno dell'ambiente di esplorazione. L'ostacolo non può essere attraversato dal drone e rappresenta la base per modellare diversi tipi di strutture come, ad esempio, alberi, costruzioni, ecc.

5.2.6. Elemento target

Un target è un elemento collocato in una posizione statica e sconosciuta a priori. Esso ricopre una singola cella dell'ambiente di ricerca e rappresenta la base per modellare diversi tipi di sostanze da rilevare. Ad esempio, un gas o un liquido tossico può essere modellato attraverso una distribuzione gaussiana di target, con media in corrispondenza della sorgente; invece, una mina anti-uomo può essere modellata come una singola cella.

Lo stato di un target è identificato dalla variabile *target.state*, che può assumere i seguenti valori:

- Found;
- Not Found.

5.2.6.1. Algoritmo di rilevamento del target

Come già evidenziato nella fase di analisi, il drone è equipaggiato con uno o più sensori. Si suppone che un sensore sia ideale e la sua copertura sia modellata attraverso un cono di sensing, con un angolo e un raggio definiti a priori.

Durante l'esplorazione, il drone può rilevare o nessun target o un solo target oppure più di un target. Nel primo caso il drone prosegue normalmente la sua esplorazione. Nel secondo caso, il drone rilascia un'impronta di feromone centrata sul target rilevato: a seconda della lunghezza del raggio di sensing, il feromone può essere rilasciato più o meno vicino alla posizione corrente del drone. Nell'ultimo caso, invece di rilasciare più impronte di feromone, il drone ne rilascia una sola in corrispondenza del target più vicino (come evidenziato nella figura 57).

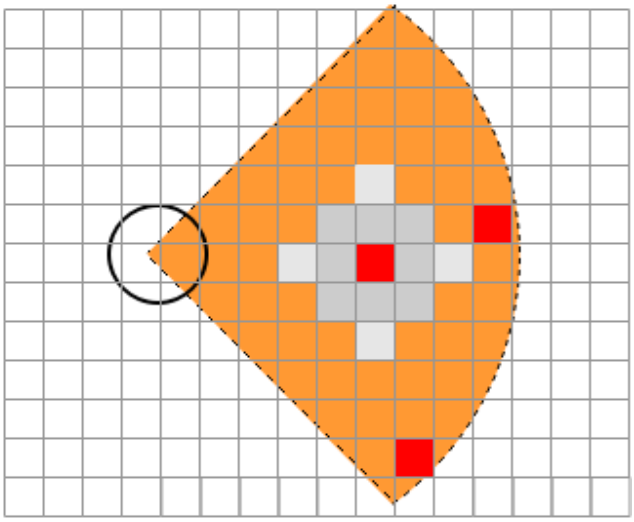

Figura 25 - Rilascio del feromone nel caso di rilevamento di più di un target

Adottando tale strategia, il drone rilascerà le impronte di feromone sugli altri target nei ticks immediatamente successivi.

Lo stato di un target rilevato viene contrassegnato come "found". Il seguente diagramma di attività illustra il funzionamento dell'algoritmo di rilevamento del target.

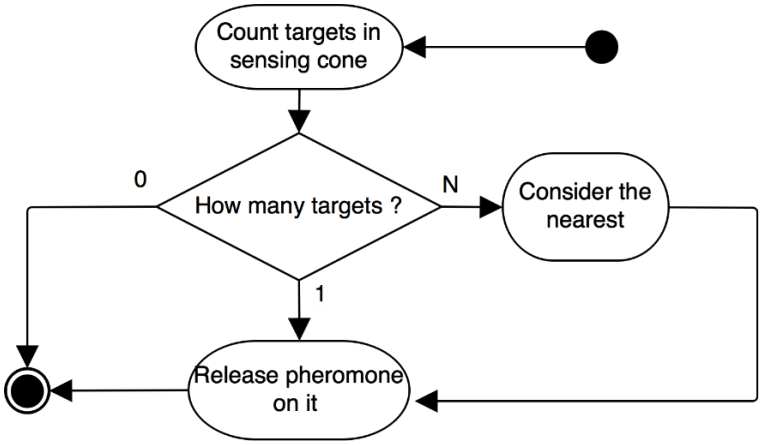

Figura 26 - Diagramma di attività del funzionamento dell'algoritmo di rilevamento del target

5.2.6.2. Diagramma di classe dell'elemento target

La classe target è una classe derivata dalla classe patch, perché un target è fondamentalmente una patch con informazioni di stato aggiuntive.

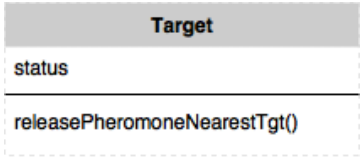

Figura 27 - Diagramma di classe elemento target

5.2.7. Logica del drone

Dopo aver progettato e descritto ciascun modulo software del simulatore SciaDro, risulta necessario stabilire un coordinamento tra i diversi moduli. Il diagramma di attività presentato nella figura 60 illustra la logica con cui i moduli software sono coordinati, al fine di soddisfare i requisiti. L'insieme delle attività eseguite dal drone è definita "*logica del drone*".

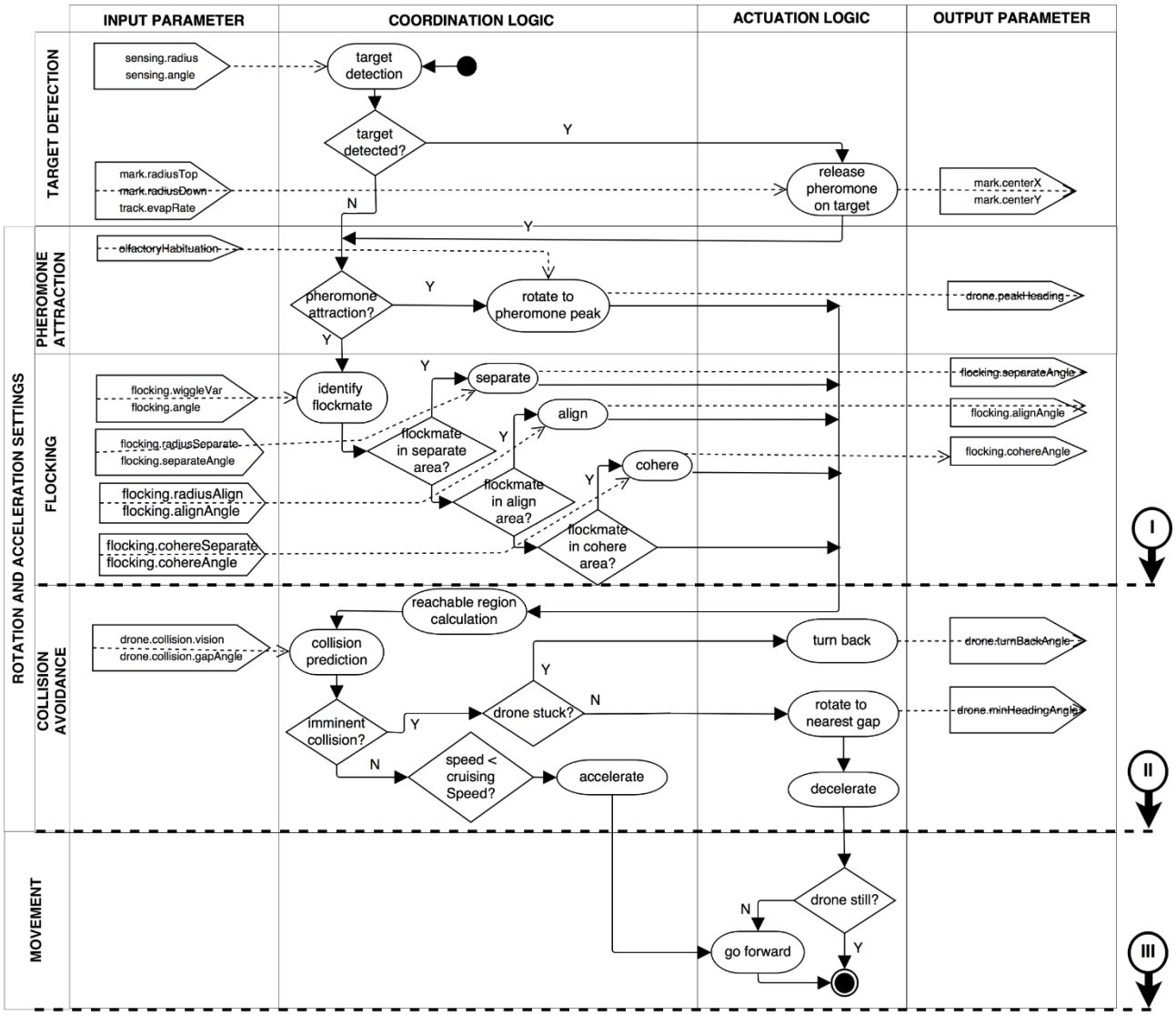

Figura 28 - Logica del drone

Come evidenziato dal diagramma di attività, la logica del drone è suddivisa in tre fasi. All'interno di ogni fase, il drone deve eseguire tutte le operazioni previste dalla fase stessa. L'ordine delle fasi è dovuto al fatto che il meccanismo di collision avoidance ha una priorità più alta rispetto al meccanismo di coordinamento basato su flocking e stigmergia.

La prima fase consiste di tre operazioni: il rilevamento del target, l'attrazione del feromone e il flocking. Il rilevamento del target viene eseguito in tutti i casi. L'attrazione del feromone viene eseguita in caso di presenza di una traccia stigmergica e comporta una rotazione verso il picco di intensità del feromone. Il

flocking viene eseguito in caso di assenza di attrazione del feromone e comporta l'esecuzione di una delle seguenti operazioni: separazione, allineamento, coesione o nessuna delle tre.

La seconda fase è rappresentata dal meccanismo di collision avoidance. Ciascun drone schedula un cono di patches al fine di informare gli altri droni sulle possibili posizioni che può ricoprire nell'immediato futuro. Il meccanismo di collision avoidance è applicato da tutti i droni dello sciame. Il processo complessivo è denominato "calcolo della regione raggiungibile" (reachable region). Le posizioni già schedulate sono trattate come ostacoli dagli altri droni. La seconda e la terza fase sono state divise perché, prima di eseguire un movimento, ciascun drone deve schedulare la sua posizione futura.

Infine, la terza fase è rappresentata dal movimento. A seconda delle posizioni relative degli ostacoli, il drone esegue un movimento seguendo la logica prevista dal meccanismo di collision avoidance.

5.2.7.1. Diagramma delle classi

La maggior parte delle classi descritte in precedenza risultano collegate da relazioni di vario tipo.

La classe drone è la classe principale. Il drone è equipaggiato con uno o più sensori e rilascia le impronte di feromone. Inoltre, sfrutta il meccanismo di flocking per esplorare l'ambiente in modo efficiente e il meccanismo di collision avoidance per evitare gli ostacoli e gli altri droni. I meccanismi di flocking e stigmergia possono essere abilitati in modo facoltativo. In questo modo, risulta possibile valutare le differenze di performance nei diversi casi possibili.

La patch è un blocco fondamentale dell'ambiente di simulazione: infatti, il target, gli ostacoli e le tracce di feromone sono costruiti a partire da questo elemento. Di conseguenza, la classe target può essere considerata una classe ereditata dalla classe patch, mentre una traccia di feromone risulta essere un'aggregazione di patches.

La figura seguente illustra le classi che compongono il software e ne evidenzia le relazioni che le collegano.

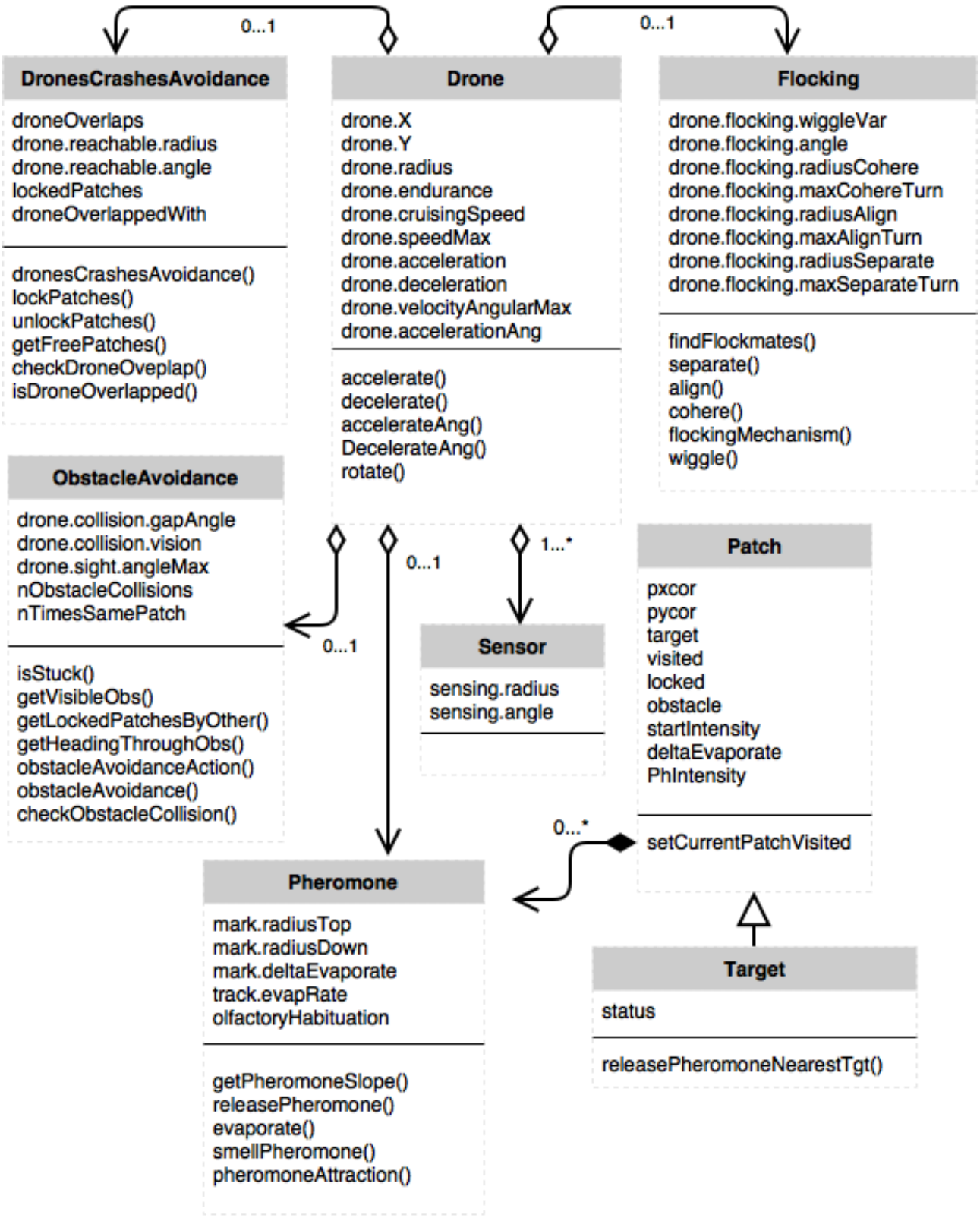

Figura 29 - Diagramma delle classi completo

5.2.8. Parametri del simulatore

La seguente tabella riassume i parametri di input del simulatore, individuati durante la fase di progettazione:

Tipo dei Paramteri	Parametro	Significato	Unità di misura
Simulazione	Strategy	La strategia da usare durante la ricerca: o flocking o stigmeriga o entrambi	-
	Number of drones	Il numero totale di droni nello sciame	-
Drone	Drone radius	Raggio físico del drone	(m)
	Cruising speed	Velocità di crociera del drone	(m/s)
	Max speed	Velocità massima	(m/s)
	Acceleration	Accelerazione lineare del drone	(m/s^2)
	Deceleration	Decelerazione lineare del drone	(m/s^2)
	Max angular speed	Velocità angolare massima del drone	(rad/s)
	Angular acceleration	Accelerazione angolare del drone	(rad/s^2)
	Angular deceleration	Decelerazione angolare del drone	(rad/s^2)
	Endurance	Autonomia del drone	(min)
	Sensing radius	Raggio di sensing	(m)

	Sensing angle	Angolo di sensing	(°)
Collision avoidance	Reachable radius	Raggio del cono di patches prenotate	(m)
	Reachable angle	Angolo del cono di patches prenotate	(°)
	Angle max sight	Angolo massimo di riconoscimento degli ostacoli	(°)
	Collision vision	Raggio entro cui um ostacolo viene riconosciuto come tale	(m)
	Gap angle	Angolo mínimo attraverso cui il drone può passare	(°)
Feromone	Radius top	Raggio di massima intensità dell'impronta feromonica	(m)
	Radius down	Raggio relativo all'intensità discendente dell'impronta feromonica	(m)
	Evaporation rate	Percentuale di feromone evaporato ad ogni tick	-
	Olfactory habituation	Tempo di assuefazione all'attrazione del feromone	(s)
Flocking	Angle	Angolo di visione del flock	(°)
	Wiggle var	Angolo massimo di rotazione casuale	(°)
	Radius separate	Minima distanza tra compagni di flock	(m)

Flocking (Separate)	Max separate turn	Angolo massimo di rotazione nel meccanismo di separazione	(°)
Flocking (Align)	Radius align	Raggio dell'area di allineamento	(m)
	Max align turn	Angolo massimo di rotazione nel meccanismo di allineamento	(°)
Flocking (Cohere)	Radius cohere	Raggio dell'area di coesione	(m)
	Max cohere turn	Angolo massimo di rotazione nel meccanismo di coesione	(°)

Tabella 6 - Tabella riassuntiva dei parametri strutturali del simulatore

I parametri del simulatore possono essere configurati all'interno di un file di configurazione. Ad esempio, i parametri del drone devono essere configurati in funzione del drone selezionato per la missione. I parametri di sensing, invece, dipendono dal tipo di sensore con cui il drone è equipaggiato ovvero dal contesto applicativo.

Qualora si scelga una strategia che esclude il coordinamento basato su flocking e stigmergia, oltre ai parametri del drone, gli unici parametri a dover essere configurati sono quelli collegati al meccanismo di collision avoidance.

Un parametro importante è rappresentato dal numero di droni da usare nella simulazione. Da un lato, un numero basso non consente di completare il rilevamento dei target entro il tempo di autonomia dei droni, ma ne riduce sensibilmente la probabilità di collisione. Dall'altro lato, un numero elevato di droni esplora meglio e in meno tempo lo spazio di ricerca, ma aumenta la probabilità di collisione tra i droni.

5.2.9. Valutazione della performance

L'obiettivo della valutazione della performance è quello di stabilire la migliore configurazione dei parametri per un dato contesto applicativo. Infatti, data l'eterogeneità degli scenari in termini di numero e distribuzione di target e ostacoli, non è possibile stabilire una configurazione di parametri migliore in assoluto.

L'obiettivo principale dello sciame di droni è quello di rilevare, entro il tempo di autonomia, i target presenti all'interno del contesto applicativo. Di conseguenza, la performance può essere valutata considerando il tempo impiegato per il rilevamento di almeno il 95% dei target (*discovering time*). Una migliore configurazione dei parametri di ingresso, relativi al coordinamento basato su flocking e stigmergia, comporterà un discovering time inferiore.

Un altro parametro per la valutazione della performance è rappresentato dal numero di collisioni. Questo parametro risulta importante per due ragioni fondamentali:

1. Costi: in un contesto reale, la perdita di un drone può avere un impatto economico più o meno rilevante a seconda del tipo di drone;
2. Rischi: la collisione di un drone può compromettere l'incolumità di persone o cose.

Al fine di ottenere un risultato significativo da un punto di vista statistico, la performance di un insieme di parametri di ingresso viene valutata come media su più simulazioni indipendenti.

5.2.10. Ottimizzazione basata su Differential Evolution

L'algoritmo Differential Evolution è uno strumento software in grado di trovare la configurazione ottimale dei parametri per un problema caratterizzato da una specifica complessità. E' importante sottolineare che l'algoritmo Differential Evolution non trova la soluzione ottima del problema, ma una soluzione ottimale, ovvero una soluzione "vicina" a quella ottima.

I parametri relativi al flocking e alla stigmergia sono gli unici ad essere configurati attraverso lo strumento Differential Evolution. Infatti, l'obiettivo principale del simulatore SciaDro è quello di valutare l'impatto dei meccanismi di coordinamento basati su flocking e stigmergia sui diversi scenari applicativi. Inoltre, si intende confrontare la soluzione ottimale relativa alla nuova versione del simulatore (ver. 2.0) con quella relativa alla versione precedente (ver. 1.2).

L'algoritmo DE ricerca il valore ottimale per ciascun parametro di configurazione all'interno di un intervallo definito a priori. Gli estremi di tale intervallo devono essere scelti in modo accurato in relazione alle caratteristiche del contesto applicativo e alle specifiche reali dei droni e dei sensori di

rilevamento. Ad esempio, un angolo di visione del flock di 180° è idoneo da un punto di vista tecnologico. Nel caso del confronto con la versione precedente del simulatore i valori degli intervalli possono essere scelti in base ai corrispondenti valori dei parametri usati nella versione precedente.

5.2.10.1. Funzionamento dell'algoritmo Differential Evolution

Il Differential Evolution è un algoritmo evolutivo il cui scopo è quello di trovare delle soluzioni candidate ottime rispetto ad una misura di qualità in modo iterativo. Questo algoritmo è un metodo di ottimizzazione capace di gestire correttamente funzioni non differenziabili e non lineari. La ricerca del valore ottimo inizia con una popolazione generata in modo casuale, dove i valori assunti da ciascun elemento sono vincolati ad intervalli scelti a priori. Come molti algoritmi che rientrano in questa categoria, opera con tre operazioni fondamentali: mutazione, crossover, selezione. Lo spazio delle soluzioni viene esplorato attraverso dei vettori di parametri generati per differenza; questo comportamento si individua nell'operazione di mutazione.

La maggior differenza fra un algoritmo genetico ed uno evolutivo sta nel fatto che il primo dipende dall'operazione di crossover (rimescolamento di geni), mentre l'altro utilizza l'operazione di mutazione come primo meccanismo di ricerca.

Il Differential Evolution utilizza delle differenze pesate fra i vettori delle soluzioni al fine di perturbare la popolazione, inoltre non richiede la codifica binaria dei membri della popolazione.

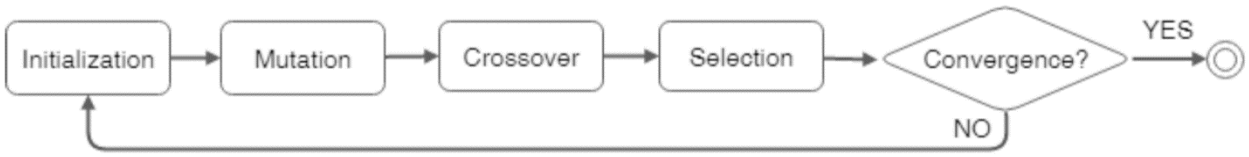

Figura 30 - Passi dell'algoritmo Differential Evolution

Si riporta lo pseudo codice del Differential Evolution e si descrivono con maggior rigore le tre operazioni fondamentali dell'algoritmo.

```
Begin
  Generate randomly an initial population of solutions.
  Calculate the fitness of the initial population.
  Repeat
    For each parent, select three solutions at random.
    Create one offspring using the DE operators.
    Do this a number of times equal to the population size.
    For each member of the next generation
      If offspring(x) is more fit than parent(x)
        Parent(x) is replaced.
  Until a stop condition is satisfied.
End
```

5.2.10.1.1. Mutazione

La mutazione viene utilizzata per produrre una popolazione di Np vettori; questa operazione genera un nuovo vettore mutante, aggiungendo una frazione della differenza, fra due vettori selezionati casualmente, ad un terzo. L'implementazione elementare di questa operazione è descritta dalla formula qui riportata:

$$v_i = x_{r0} + F(x_{r1} - x_{r2})$$

dove v_i è il nuovo vettore mutante, mentre i vettori x_{rj} sono i tre vettori selezionati a caso tra quelli disponibili. La funzione F ε (0,1+) controlla il tasso con cui la popolazione evolve.

5.2.10.1.2. Crossover

Quest'operazione produce i vettori da testare a partire dai vettori dei parametri. In particolare, viene incrociato ogni vettore della popolazione corrente con un vettore mutante. L'operazione di crossover è descritta nella formula seguente:

$$u_i = \begin{cases} v_{i,j} & if(rand_j(0,1) \leq Cr) \\ u_{i,j} & otherwise \end{cases}$$

Il nuovo individuo viene creato scegliendo, tra il vettore mutante e il corrente con una probabilità Cr. Questa probabilità è chiamata probabilità di crossover ed è un parametro definito dall'utente.

5.2.10.1.3. Selezione

Se il vettore della soluzione candidata in uso, u_i, ottiene un valore della funzione obbiettivo minore o uguale rispetto alla soluzione candidata migliore, la migliore soluzione viene sostituita dal vettore u_i nella generazione successiva.

$$x_{i,g+1} = \begin{cases} u_{i,g} & if(f(u_{i,g}) \leq f(x_{i,g})) \\ x_{i,g} & otherwise \end{cases}$$