



Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica
e Comunicazione
Corso di laurea in Informatica

Sistemi Complessi

School of Fish Behaviour and

Predator

Minelli Monica 793477
Nastasi Alessandro 793025
Rizzi Luca 793971

Luglio 2018

Indice

1	Introduzione	5
1.1	Struttura della relazione	5
2	NetLogo	6
3	Ambiente	7
4	Agenti: Pesci Preda	8
4.1	Aree del Pesce Preda	8
4.1.1	Area di Visibilità e Interazione con Predatore	8
4.1.2	Area di Interazione con gli altri Pesci Preda	10
4.2	Comportamenti del Pesce Preda	11
4.2.1	Schooling	11
4.2.1.1	Direzione	12
4.2.1.2	Velocità	13
4.2.2	Cooperative Escape	13
4.2.2.1	Direzione	13
4.2.2.2	Velocità	14
4.2.3	Selfish Escape	14
4.2.3.1	Direzione	14
4.2.3.2	Velocità	14
5	Agenti: Pesci Predatori	15
5.1	Area del Pesce Predatore	15
5.2	Comportamenti del Pesce Predatore	16
5.2.1	Movimento Normale	16
5.2.1.1	Direzione	16
5.2.1.2	Velocità	16
5.2.2	Movimento verso i Pesci Preda	17
5.2.2.1	Direzione	17
5.2.2.2	Velocità	17
5.2.3	Attacco	17
5.2.3.1	Direzione	17
5.2.3.2	Velocità	17
6	Assunzioni	18
7	Simulazioni	19
7.1	Parametri Utilizzati	19
7.2	Risultati Simulazione	19
7.3	Face Validation	21
8	Analisi di sensitività	22
8.1	Analisi con 1 predatore	23
8.2	Analisi con 5 predatori	24
8.3	Analisi con 10 predatori	25
8.4	Modifica Dimensione Ambiente	26
8.5	Considerazioni	27

9 Conclusioni	28
A Implementazione	29
A.1 Variabili	29
A.2 Funzioni	30
B Listato del codice	32

Elenco delle figure

1	Interfaccia NetLogo	6
2	Parametri Ambiente Netlogo	7
3	Aree di Interazione con il Predatore	8
4	Regole di scelta del CBP	11
5	Aree di Interazione con le Prede	15
6	Esempio di esecuzione con 800 prede e 1 predatore	20

1 Introduzione

La capacità degli animali di unirsi in gruppo ha ispirato molti ricercatori di diverse discipline per descrivere, modellare e riprodurre questi fenomeni [5].

La maggior parte degli animali che vivono in gruppo sono in grado di muoversi coerentemente e collettivamente, preservando caratteristiche comuni come le svolte coordinate, il mantenimento delle strutture interne e il movimento apparentemente senza leader.[1]

La teoria della complessità indica che vaste popolazioni di unità possono auto-organizzarsi in aggregazioni che generano schemi, immagazzinano informazioni e si impegnano in decisioni collettive. [4]

Le aggregazioni spesso si comportano come una unità con proprietà che non sono semplicemente una somma dei comportamenti individuali.

In biologia l'aggregazione di pesci è un termine generale che identifica qualsiasi collezione di pesci che si aggrega nella stessa località.

Si parla di banco di pesci nel momento in cui l'aggregazione è strutturata, i componenti sono della stessa specie e della stessa dimensione e età. I banchi di pesci si muovono in coordinazione, ogni singolo individuo è situato allo stesso spazio l'uno dall'altro.

La funzione primaria del banco di pesci è la protezione dei membri in caso di attacchi da parte di predatori.

Studi teorici ed empirici[3] dimostrano che per un individuo il rischio è minore nel caso di gruppi coesi e grandi, grazie soprattutto agli effetti di confusione.

Per implementare il modello illustrato successivamente, abbiamo pensato di basarci sulla teoria vista a lezione, principalmente sul concetto di *boids*; infatti questa tipologia di agenti hanno tre comportamenti tipici molto utili per descrivere movimenti di singoli individui in gruppo: *separazione*, *coesione* e *allineamento* sono le componenti principali che anche nel nostro progetto, seppur con modifiche, verranno utilizzate.

1.1 Struttura della relazione

- Nella **Sezione 2** verrà proposta una breve introduzione sullo strumento utilizzato per l'implementazione;
- Le **Sezioni 3, 4 e 5** descriveranno dal punto di vista matematico e teorico il modello dell'ambiente e degli agenti che abbiamo implementato;
- Nella **Sezione 6** sono presenti le assunzioni fatte in fase di modellazione;
- La **Sezione 7** mostrerà un esempio di simulazione;
- La **Sezione 8** presenterà i risultati ottenuti dalle simulazioni al variare di alcuni parametri fondamentali;
- La **Sezione 9** presenterà alcune considerazioni conclusive sul lavoro svolto.

2 NetLogo

Per il nostro progetto è stato scelto un ambiente di modellazione programmabile multi-agente, NetLogo ¹. Tramite esso è stato possibile modellare l'ambiente, il mare, e i boids, pesci prede e pesci predatori. Sono stati creati tramite codice gli elementi interessati nel modello assegnando ad ognuno gli attributi che li caratterizzava.

Per l'interfaccia (Figura 8a) di visualizzazione del tool sono stati invece inseriti degli slider per modificare con più facilità alcuni attributi dei pesci e controllare i differenti comportamenti a seconda delle impostazioni, e due slider per poter scegliere quanti pesci prede e quanti pesci predatori inserire nel modello. Sono stati aggiunti due bottoni, uno per inizializzare il modello secondo le impostazioni scelte, e il secondo per permettere di iniziare la computazione secondo le regole scritte nel codice. Nell'interfaccia è stata aggiunta la visualizzazione dell'ambiente nel quale è possibile vedere la simulazione del movimento dei pesci e delle regole implementate. L'ultimo elemento che è stato aggiunto nell'interfaccia è un grafico con titolo "Number of Living Preys" che permette di vedere quanti pesci restano in vita (asse delle ordinate) al passare del tempo, ticks (asse delle ascisse).

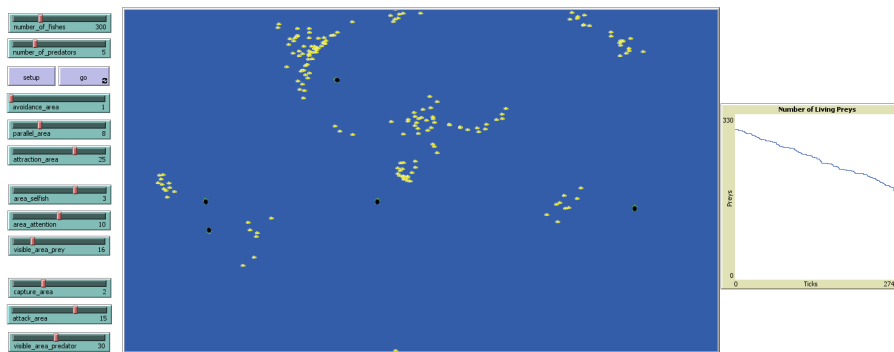


Figura 1: Interfaccia NetLogo

¹<https://ccl.northwestern.edu/netlogo/>

3 Ambiente

Per rendere la modellazione il più possibile vicina alla realtà abbiamo deciso di adottare un ambiente continuo. Concettualmente uno spazio è considerato continuo se contiene infiniti elementi e tra essi non ci sono *spazi vuoti*. Nel nostro caso ciò implica che gli agenti possano assumere qualsiasi posizione arbitraria. L'ambiente sarà toroidale, cioè i bordi si uniranno, le caselle dei bordi di destra saranno uniti a quelli dei bordi di sinistra e quelle dei bordi superiori a quelle dei bordi inferiori. Tutti gli agenti sono punti e hanno una proprietà posizione che ne salva le coordinate x e y come una tupla.

Avendo utilizzato Netlogo per l'implementazione, la creazione dell'ambiente è stata effettuata modificando i parametri nella sezione *Settings* del framework. Nello specifico, per visualizzare tutti i componenti dell'interfaccia grafica e avere un ambiente abbastanza grande da poter osservare i differenti comportamenti degli agenti, i parametri impostati sono stati quelli presenti nella figura sottostante.

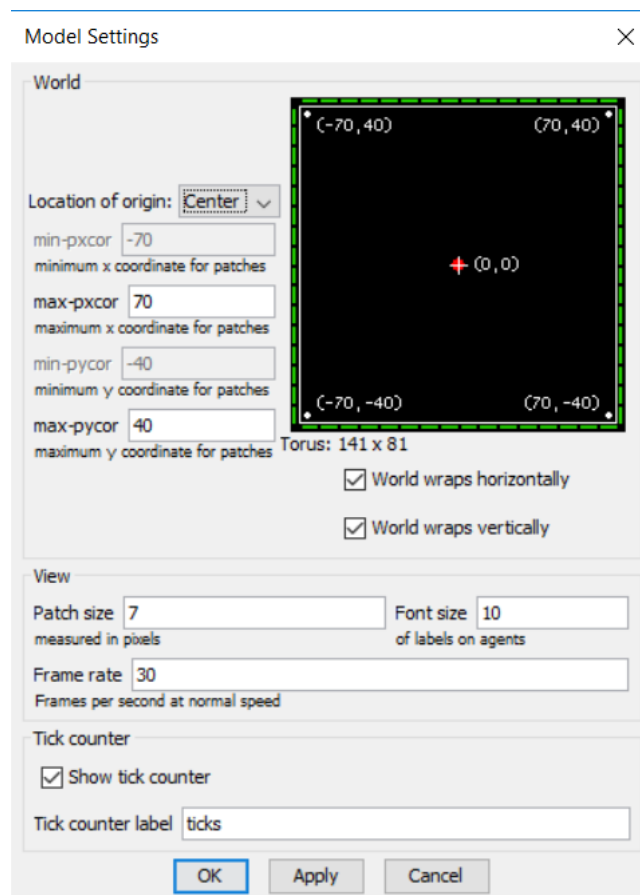


Figura 2: Parametri Ambiente Netlogo

4 Agenti: Pesci Preda

Il pesce preda sarà l'agente più complicato da modellare in quanto avrà molti comportamenti differenti e oltre a dover considerare l'interazione con il predatore, dovrà essere modellata anche l'interazione con altre prede per formare i banchi.

Come anticipato precedentemente gli agenti verranno modellati come dei boids. I boids rientrano nella classificazione di agenti tropistici.

L'**agente tropistico** è definito da una tupla di sei elementi:

$$\langle E, P, A, see, do, action \rangle$$

dove:

- E : corrisponde al set degli stati dell'ambiente
- P : sottoinsieme di E , set degli stati visti dall'agente
- A : corrisponde al set di azioni
- $see: A \rightarrow P$
- $do: P \rightarrow A$
- $action: A \times E \rightarrow E$

4.1 Aree del Pesce Preda

Le aree che andremo a considerare successivamente saranno indispensabili per la modellazione dei comportamenti e quindi per descrivere la dinamica di movimento e di interazione con gli altri agenti.

4.1.1 Area di Visibilità e Interazione con Predatore

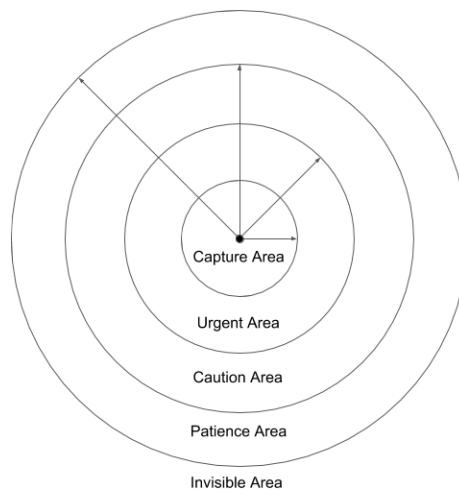


Figura 3: Aree di Interazione con il Predatore

Ogni agente ha 4 raggi che identificano le aree nel suo intorno per l'interazione con il predatore[2].

I raggi definiti sono i seguenti:

- $r_{capture}$: è il raggio che definisce l'area di cattura.
- r_{urgent} : è il raggio che definisce l'area di urgenza.
- $r_{caution}$: è il raggio che definisce l'area in cui il pesce deve prestare attenzione.
- $r_{patience}$: è il raggio che definisce l'area in cui il pesce può portare pazienza.

I raggi definiti in precedenza danno origine alle aree di interazione con il predatore; con R_{pp} indicheremo la distanza tra la preda *i-esima* e il predatore.

- *Capture area*: è l'area in cui, in presenza di predatore, la preda viene catturata.

$$0 < R_{pp} \leq r_{capture}$$

- *Urgent area*: è l'area in cui, in presenza di predatore, il pesce modifica il proprio comportamento cercando solamente di fuggire per salvarsi.

$$r_{capture} < R_{pp} \leq r_{urgent}$$

- *Caution area*: è l'area in cui, in presenza di predatore, la preda modifica il proprio comportamento attuando una strategia di cooperative escape.

$$r_{urgent} < R_{pp} \leq r_{caution}$$

- *Patience area*: è l'area in cui, in presenza di predatore, la preda non modifica il proprio comportamento e continua a rimanere in branco (schooling).

$$r_{caution} < R_{pp} \leq r_{patience}$$

- *Invisible area*: è l'area in cui il pesce non vede.

$$R_{pp} > r_{patience}$$

I casi presi in considerazione precedentemente valgono per la singola preda che si trova già in un banco. Nel caso in cui la preda sia ancora da sola nel momento in cui un predatore entra nella sua area visibile, la probabilità di essere catturata aumenta, in quanto essa applicherà solamente il meccanismo di selfish escape nel momento in cui il predatore sarà già troppo vicino; siccome la velocità del predatore aumenta nel momento in cui sferra l'attacco, la preda se presa singolarmente verrà eliminata.

4.1.2 Area di Interazione con gli altri Pesci Preda

Le prede devono inoltre tenere conto della distanza con altre prede; per fare ciò avranno altri 3 raggi:

- $r_{avoidance}$: è il raggio che definisce l'area in cui, in presenza di un'altra preda, il soggetto deve cercare di allontanarsi per evitare la collisione.
- $r_{parallel}$: è il raggio che definisce l'area in cui, in presenza di un'altra preda, il soggetto modifica il proprio movimento per cercare di allinearsi al movimento del/i vicino/i.
- $r_{attraction}$: è il raggio che definisce l'area in cui, in presenza di un'altra preda, il soggetto modificherà il proprio movimento per cercare di avvicinare il vicino.

I raggi definiti precedentemente, danno origine ad aree di interazione tra le varie prede; con R_{ij} definiamo la distanza tra la preda i -esima e la preda j -esima.

- *Area di avoidance*: è l'area in cui, in presenza di un'altra preda, il soggetto cerca di allontanarsi da essa per evitare collisioni.

$$0 < R_{ij} \leq r_{avoid}$$

- *Area di parallel orientation*: è l'area in cui, in presenza di un'altra preda, il soggetto allinea il proprio movimento ad quello del vicino.

$$r_{avoid} < R_{ij} \leq r_{parallel}$$

- *Area di attrazione*: è l'area in cui, in presenza di un'altra preda, il soggetto modifica il proprio movimento per avvicinare il vicino.

$$r_{parallel} < R_{ij} \leq r_{attraction}$$

Nel caso in cui nessun'altra preda sia presente nelle aree definite in precedenza di un soggetto, esso sceglierà la propria direzione in modo randomico.

4.2 Comportamenti del Pesce Preda

La figura 4 mostra la scelta che effettuano le prede in base alla presenza di un predatore all'interno delle aree definite in precedenza. I CBP (Component Behavior Pattern) verranno spiegati nello specifico nelle sezioni successive, definendo in modo preciso l'implementazione che è stata fatta.

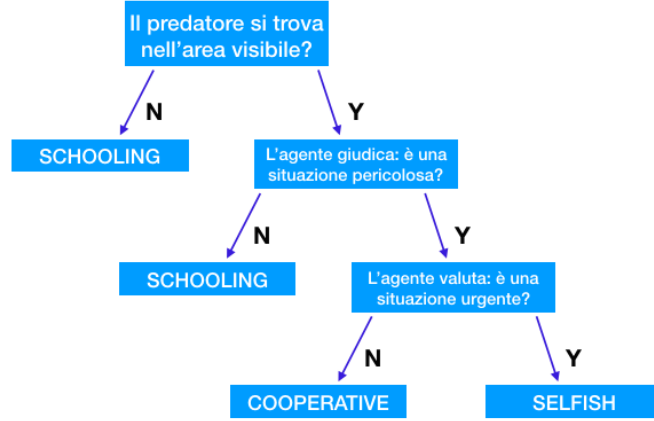


Figura 4: Regole di scelta del CBP

Il singolo pesce *i-esimo* si può muovere seguendo la formula:

$$\begin{aligned}
 x_i(t + \Delta t) &= x_i(t) + \Delta t * v_i(t + \Delta t) * \cos \alpha_i(t + \Delta t) \\
 y_i(t + \Delta t) &= y_i(t) + \Delta t * v_i(t + \Delta t) * \sin \alpha_i(t + \Delta t)
 \end{aligned}$$

dove $v_i(t + \Delta t)$ indica la velocità di movimento del pesce *i-esimo* e $\alpha_i(t + \Delta t)$ la direzione del movimento. Il processo di movimento consiste in due passaggi: il primo quando il pesce cambia la direzione del movimento, il secondo quando aumenta o diminuisce la propria velocità. A seconda della scelta dell'azione (Schooling - Selfish Escape - Cooperative Escape) il valore di $\alpha_i(t + \Delta t)$ cambierà. In Netlogo questa formula è implementata in modo simile nei metodi di movimento verso (*towards*) un agente.

4.2.1 Schooling

Viene considerata l'**allelomimesi** (*"doing what your neighbors do"*) come metodo principale per il mantenimento del banco. La condizione di base è che ogni individuo conosca solamente una situazione locale intorno a sé. I movimenti essenziali per la generazione del banco consistono in:

- Evitare collisioni con i vicini
- Muoversi in parallelo con i vicini
- Cercare di rimanere all'interno del banco
- Avvicinarsi ai vicini

Questi movimenti vengono controllati usando le aree di interazioni tra prede descritti in precedenza.

4.2.1.1 Direzione

Per studiare la direzione abbiamo analizzato questo paper[7]. La nuova direzione di movimento è data da:

$$\alpha_i(t) + [\gamma_{AL}(i)(\beta_i^{AL}(t + \Delta t) - \alpha_i(t)) + \gamma_{AV}(i)(\beta_i^{AV}(t + \Delta t) - \alpha_i(t))]$$

dove: $\gamma_{AL}(i)$ e $\gamma_{AV}(i)$ sono rispettivamente i coefficienti di contribuzione del movimento allelomimetico e della collision avoidance; inoltre deve valere la seguente condizione $\gamma_{AL}(i) + \gamma_{AV}(i) = 1$.

Il movimento dell'agente deve tenere simultaneamente in considerazione le direzioni $\beta_i^{AL}(t + \Delta t)$ e $\beta_i^{AV}(t + \Delta t)$. La prima direzione è determinata nel modo seguente:

- Quando i vicini *j-esimi* sono nell'area di orientamento parallelo della preda *i-esima*, essa modificherà la propria direzione per girarsi nella stessa direzione della media delle direzioni delle prede *j-esime*.

$$\beta_i^{AL}(t + \Delta t) = \frac{\sum_{j=0}^k \alpha_j(t)}{k}$$

dove k indica il numero massimo di pesce preda nell'intorno.

- Quando il vicino *j* è nell'area di attrazione, il pesce *i* si muove nella direzione del pesce *j* (che è il più vicino di quelli nell'area di attrazione) per avvicinarsi ad esso.

$$\cos \beta_i^{AL}(t + \Delta t) = \frac{x_j(t) - x_i(t)}{\sqrt{(x_j(t) - x_i(t))^2 + (y_j(t) - y_i(t))^2}}$$

$$\beta_i^{AL}(t + \Delta t) = \arccos(\cos \beta_i^{AL}(t + \Delta t))$$

- Se la preda si muove in parallelo, vuol dire che ha almeno un vicino nell'area in cui dovrà cercare di mantenere la direzione del banco, quindi cercherà, oltre a muoversi in parallelo alla media delle direzione dei vicini, di rimanere coeso con il gruppo; per fare ciò:

$$\beta_i(t + \Delta t) = \frac{\sum_{j=0}^k DR(i, j; t)}{k}$$

sarà cioè la media delle direzioni di tutti i vicini nell'area di movimento parallelo rispetto alla preda *i-esima*.

- Quando non ci sono pesci in nessuna delle due aree la preda *i* cambia randomicamente la propria direzione di movimento per cercare un pesce di cui mimare i movimenti.

$$\beta_i^{AL}(t + \Delta t) = \alpha_i(t) + \text{random}(360^\circ)$$

La seconda direzione è determinata cercando di evitare di collidere con un altro pesce nell'area di avoidance in questo modo:

$$\beta_i^{AV}(t + \Delta t) = DR(i, j; t) + 180^\circ + \text{random}(5^\circ)$$

dove $DR(i, j; t)$ è la direzione dal pesce i alla preda più vicina j -esima al tempo t , sommando 180° manda la preda nella direzione opposta.

4.2.1.2 Velocità

Per quanto riguarda la velocità, essa verrà considerata in questo modo:

$$v_i(t + \Delta t) = v_i^{fix}(t + \Delta t) + \text{rumoreGaussiano}$$

dove $v_i^{fix}(t + \Delta t)$ indica la velocità assegnata al pesce preda e il *rumoreGaussiano* è un valore estratto da $N(\mu, \sigma^2)$, in modo da perturbare la velocità e non avere agenti che si muovano come un automa.

4.2.2 Cooperative Escape

Il meccanismo di cooperative escape viene messo in atto nel momento in cui il predatore non è ancora troppo vicino da essere un pericolo imminente ma è comunque nell'area in cui la preda deve iniziare a preoccuparsi di questa presenza. Siccome, come detto in precedenza, se la preda venisse isolata sarebbe mangiata molto facilmente, in questo comportamento l'obiettivo è tenere in considerazione sia il movimento ad allontanarsi dal predatore, sia il movimento necessario a rimanere il più possibile in banco.

4.2.2.1 Direzione

La nuova direzione di movimento è data da:

$$\alpha_i(t + \Delta t) = \alpha_i(t) + [\gamma_{SC}(i)(\beta_{SC}(i; t + \Delta t) - \alpha_i(t)) + \gamma_{RA}(i)(\beta_{RA}(i; t + \Delta t) - \alpha_i(t))]$$

dove $\beta_{SC}(i; t + \Delta t)$ è la direzione del movimento determinato nella situazione dove la preda i prende in considerazione solo il vicino e $\gamma_{SC}(i)$ e $\gamma_{RA}(i)$ saranno i pesi che condizionano le direzioni, e dovranno soddisfare la condizione seguente: $\gamma_{SC}(i) + \gamma_{RA}(i) = 1$. La sua direzione cercherà di mantenere il comportamento di schooling. Infatti, riprendendo la formula dello schooling spiegata precedentemente, si avrà che:

$$\beta_{SC}(i; t + \Delta t) = \gamma_{AL}(i)\beta_i^{AL}(t + \Delta t) + \gamma_{AV}(i)\beta_i^{AV}(t + \Delta t)$$

Mentre $\beta_{RA}(i; t + \Delta t)$ è la direzione del movimento determinato nella situazione dove la preda i prende in considerazione il predatore. La sua direzione è la migliore adottabile per scappare dal predatore. Si avrà che, come verrà spiegato meglio successivamente, nel caso in cui la preda stia cercando di scappare dal predatore, la formula utilizzata sarà la seguente:

$$\beta_{RA}(i; t + \Delta t) = DR(i, pred; t) + 180^\circ$$

dove $DR(i, pred; t)$ è la direzione del pesce i misurata dalla posizione del predatore al tempo t .

In fase di implementazione, dopo aver calcolato la nuova direzione come combinazione delle formule viste in precedenza, abbiamo inserito un angolo ulteriore (abbastanza piccolo) in modo da permettere un movimento più naturale e non far muovere tutte le prede allo stesso modo.

4.2.2.2 Velocità

Per quanto riguarda la velocità, essa verrà considerata nello stesso modo in cui è stata pensata per il comportamento di schooling con una singola modifica:

$$v_i(t + \Delta t) = v_i^{fix}(t + \Delta t) + rumoreGaussiano$$

In questo caso v_i^{fix} assumerà valore pari a v_{care} , cioè la velocità assegnata alla preda nella situazione in cui è presente il predatore all'interno della propria area di attenzione.

4.2.3 Selfish Escape

Il meccanismo di selfish escape verrà messo in atto nel momento in cui il predatore sarà particolarmente vicino alla preda e quest'ultima quindi non avrà interesse nel cercare di rimanere in gruppo ma cercherà solamente di salvarsi per allontanarsi il più possibile. Per fare ciò, quindi, verranno utilizzate parte delle formule già viste nei comportamenti precedenti.

4.2.3.1 Direzione

La nuova direzione di movimento è data da:

$$\alpha_i(t + \Delta t) = DR(i, pred; t) + 180^\circ$$

dove $DR(i, pred; t)$ è la direzione del pesce i misurata dalla posizione del predatore al tempo t e sommando un angolo di 180° la preda riuscirà scappare nella direzione opposta rispetto alla direzione del predatore.

Nell'implementazione proposta abbiamo inserito la possibilità di una piccola variazione random per non avere un comportamento in cui le prede scappino esattamente tutte nella direzione opposta ma abbiano un movimento più fluido e naturale.

4.2.3.2 Velocità

La velocità verrà considerata come sempre:

$$v_i(t + \Delta t) = v_i^{fix}(t + \Delta t) + rumoreGaussiano$$

In questo caso v_i^{fix} assumerà valore pari a v_{max} , cioè la velocità massima assegnata alla preda.

5 Agenti: Pesci Predatori

La modellazione dell'agente predatore è più semplice rispetto a quella dei pesci preda perchè non sarà presente la parte di schooling; il predatore infatti sarà da solo e dovrà tenere in considerazione solamente l'interazione con le prede. Per questo motivo anche per quanto riguarda l'area che determina le azioni, ne sarà presente solamente una tipologia descritta nella sezione successiva.

La difficoltà nell'implementazione del predatore sta nell'identificazione e nella modellazione delle possibili tipologie di attacco che esso dovrà mettere in atto per assicurarsi la cattura di una preda in fase di attacco. Inoltre si dovrà tenere in considerazione la possibilità che un attacco fallisca nel caso in cui l'agente non riesca a scegliere la preda corretta da catturare.

Anche il predatore, come spiegato precedentemente per l'agente preda, è un **agente di tipo tropistico**.

5.1 Area del Pesce Predatore

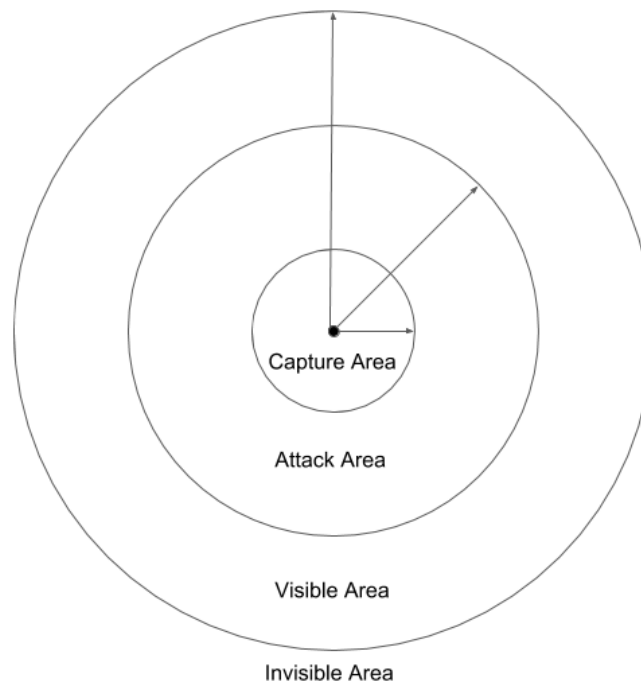


Figura 5: Aree di Interazione con le Prede

Ogni agente ha 3 raggi che identificano le aree nel suo intorno per l'interazione con le prede.

I raggi definiti sono i seguenti:

- $r_{PredCapture}$: è il raggio che definisce l'area di cattura.
- r_{attack} : è il raggio che definisce l'area di attacco.
- $r_{perception}$: è il raggio che definisce l'area in cui il predatore percepisce le prede.

I raggi descritti in precedenza identificano 3 differenti aree:

- *Area di cattura*: è l'area in cui, in presenza di prede, esse saranno mangiate dal predatore.

$$0 < R_{ji} \leq r_{PredCapture}$$

- *Area di attacco delle prede*: è l'area in cui, in presenza di prede, esso inizierà ad effettuare il proprio attacco.

$$r_{PredCapture} < R_{ji} \leq r_{attack}$$

- *Area di percezione delle prede*: è l'area in cui il predatore percepisce le possibili prede e inizia a muoversi verso esse.

$$r_{attack} < R_{ji} \leq r_{perception}$$

Con R_{ji} identificheremo la distanza tra il predatore j -esimo e la preda i -esima.

5.2 Comportamenti del Pesce Predatore

Il predatore, come detto precedentemente, avrà azioni limitate in quanto il suo obiettivo è continuare a nuotare fino a quando una preda non entra nel suo raggio di percezione. Solo a quel punto esso cercherà di muoversi verso la preda per poter sferrare un attacco nel momento in cui uno o più prede saranno nella sua area di attacco.

5.2.1 Movimento Normale

Per il movimento durante le fasi in cui non ci sono prede nell'area di visibilità, il predatore si muoverà random mantenendo una velocità costante.

5.2.1.1 Direzione

La direzione sarà determinata nel modo seguente:

$$\alpha_p(t + \Delta t) = \alpha_p(t) + random(180^\circ)$$

Nel caso del predatore, per tutti e 3 i comportamenti che potrà assumere, l'angolo di sterzata sarà inferiore rispetto a quello delle prede spiegato in precedenza. Infatti, come descritto nei vari paper analizzati, siccome la dimensione del predatore sarà superiore a quella delle prede, faranno più fatica a sterzare.

5.2.1.2 Velocità

La formula della velocità sarà la medesima rispetto a quella implementata per le prede; cambieranno, ovviamente, i valori ad essa assegnati.

$$v_p(t + \Delta t) = v_p^{base}(t + \Delta t)$$

In questo caso la velocità sarà bassa perchè il predatore non vede nei suoi dintorni alcuna preda e per questo motivo non ha alcun interesse nel muoversi velocemente.

5.2.2 Movimento verso i Pesci Preda

Il comportamento analizzato in questa sezione è quello relativo al movimento del predatore verso la preda più vicina, nel caso in cui essa non sia ancora nel suo raggio di attacco.

5.2.2.1 Direzione

La direzione in questo caso sarà determinata trovando la preda più vicina e impostando come direzione del predatore, quella che collega esso alla preda target.

$$\alpha_p(t + \Delta t) = DR(p, preda; t) + random(10^\circ)$$

$DR(p, preda; t)$ indica la direzione tra il predatore p e la preda target al tempo t . L'aggiunta dell'angolo random minore di 10° ha la funzione di permettere un movimento più naturale e che rispecchi maggiormente la realtà.

5.2.2.2 Velocità

Nel caso in cui il predatore si muova verso una preda ma essa non sia ancora nel suo raggio di attacco, esso modificherà (in modo contenuto) la propria velocità per raggiungere la preda più vicina.

$$v_p(t + \Delta t) = v_p^{mov}(t + \Delta t)$$

v_p^{mov} sarà la velocità assegnata al predatore nel momento in cui esso deciderà di muoversi per avvicinarsi a una preda.

5.2.3 Attacco

L'ultimo comportamento da implementare è la fase di attacco. In essa il predatore compie uno scatto in avanti verso la preda target per cercare di catturarla.

5.2.3.1 Direzione

Per determinare la direzione verrà utilizzata la stessa formula vista in precedenza.

$$\alpha_p(t + \Delta t) = DR(p, preda; t) + random(10^\circ)$$

$DR(p, preda; t)$ indica la direzione tra il predatore p e la preda target al tempo t . L'aggiunta dell'angolo random minore di 10° ha la funzione di permettere un movimento più naturale e che rispecchi maggiormente la realtà.

5.2.3.2 Velocità

La velocità verrà aumentata, seguendo la solita formula, per simulare un'accelerazione repentina verso la preda presa come target.

$$v_p(t + \Delta t) = v_p^{att}(t + \Delta t)$$

v_p^{att} è il valore di velocità associato alla fase di attacco.

6 Assunzioni

In questa sezione verranno analizzate alcune assunzioni fatte sul modello.

- Le prede isolate (che non fanno parte di un banco) verranno mangiate con abbastanza facilità dal predatore in quanto esso in fase di attacco ha una velocità superiore alla velocità di base della singola preda la quale, essendo da sola, non metterà in atto alcun meccanismo di fuga.
- Le prede, oltre ai meccanismi di fuga modellati, attueranno, nei confronti del predatore, un'ulteriore metodologia di fuga; esse, infatti, sfruttano il fatto che il predatore prenda come target la preda più vicina e per questo motivo riescono a far sì che esso sposti la propria attenzione molte volte in fase di attacco di un banco. Questo meccanismo è chiamato *confusione* e fornisce alle prede un'ulteriore modo di eludere gli attacchi.
- Per semplificare il modello non sono stati implementati meccanismi di riproduzione e/o nutrimento.
- In fase di implementazione abbiamo deciso di evitare la possibilità che due predatori mangino la stessa preda, creando problemi all'interno del programma.
- Abbiamo anche escluso la possibilità che due predatori possano sovrapporsi; nel caso in cui essi siano troppo vicini, si muoveranno in direzioni opposte tra di loro.
- Come spiegato nel capitolo 3, per non rendere gli agenti preda simili ad automi, per quanto riguarda la loro velocità abbiamo deciso di sommare a quella di base, un valore random preso da una distribuzione Gaussiana con $\mu = 1$ e $\sigma^2 = 0.3$. In questo modo ad ogni iterazione le prede avranno velocità differenti seppur abbastanza simili. Questa assunzione ci ha permesso di modellare la *fluttuazione* dei banchi di pesci, situazione, cioè, in cui anche all'interno dello stesso banco le prede abbiano comunque movimenti differenti.

7 Simulazioni

7.1 Parametri Utilizzati

Per la simulazione mostrata successivamente, abbiamo utilizzato il seguente set di parametri:

numero_prede	800
numero_predatori	1

avoidance_area	1
parallel_area	8
attraction_area	25

area_selfish	3
area_attention	10
visible_area_prey	16

capture_area	2
attack_area	15
visible_area_predator	30

7.2 Risultati Simulazione

Il modello creato è stato eseguito differenti volte per osservare i diversi comportamenti dei pesci a seconda del numero di predatori inseriti nell'ambiente e a seconda dei valori assegnati alle viste delle due tipologie di boids: i risultati delle differenti esecuzioni sono stati analizzati e verranno riportati nella sezione successiva dell'analisi di sensitività.

Nella figura 6 è possibile vedere alcuni frame catturati durante un'esecuzione. In questo esempio sono stati settati un grande numero di prede per poter vedere come essi si uniscano in pochi ticks in banco e come cerchino di evitare gli attacchi del predatore inserito nel modello. Nella figura 6a si può osservare come inizialmente i pesci siano disposti nell'ambiente in modo casuale e col passare del tempo (ticks) essi creano un banco (figure 6b, 6c, 6d) per difendersi dagli attacchi del predatore, cercando di confonderlo e cambiare la loro direzione (figure 6i, 6j)

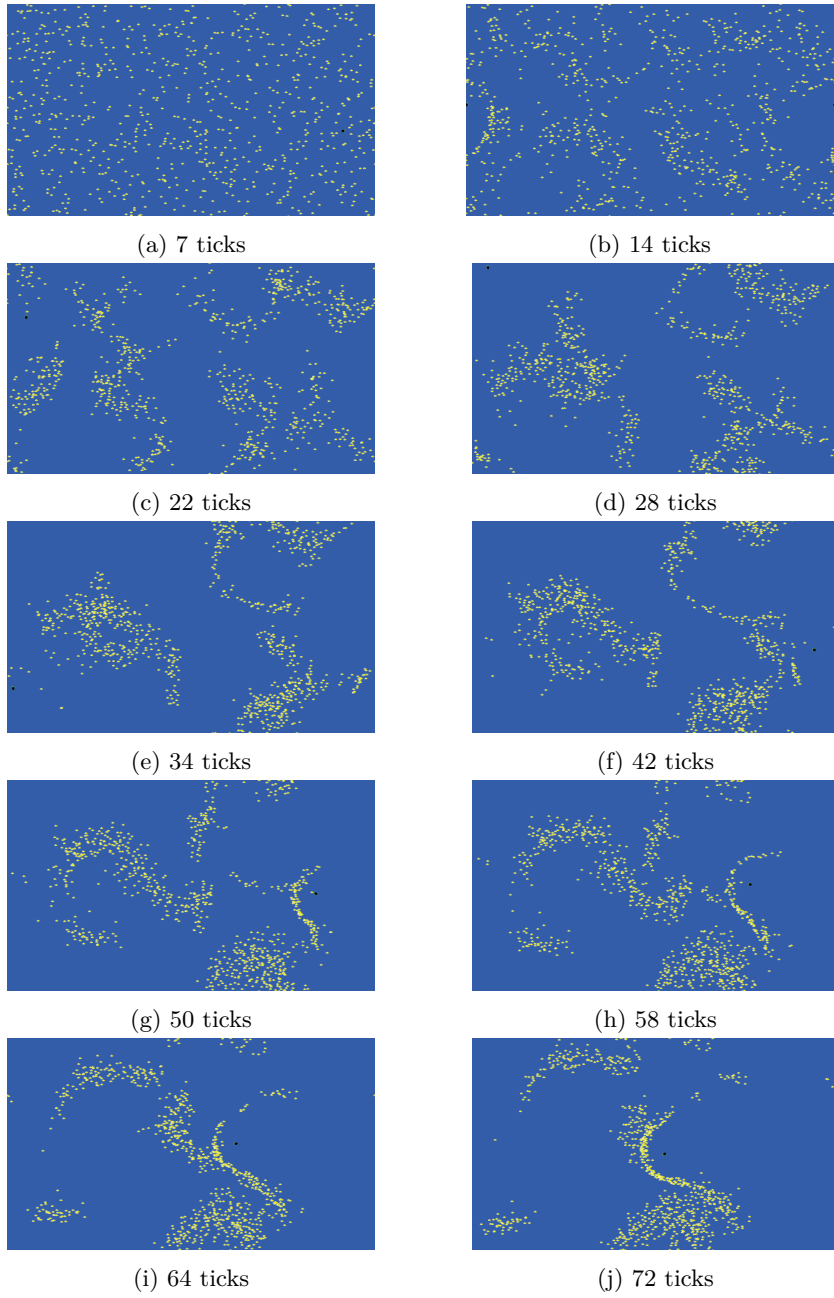
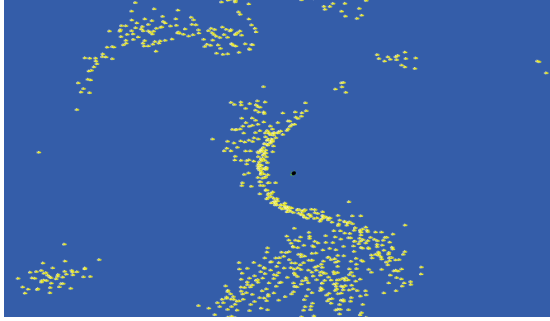


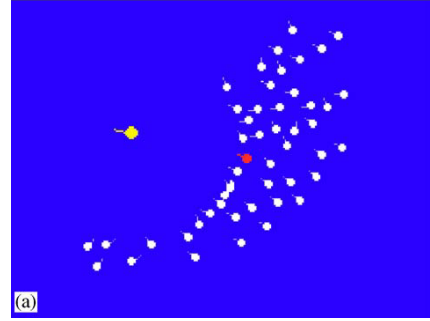
Figura 6: Esempio di esecuzione con 800 prede e 1 predatore

7.3 Face Validation

Come conferma della bontà della simulazione, almeno da un punto di vista qualitativo, il nostro risultato verrà confrontato con quello proposto dagli autori [6] del paper preso in esame.



(a) Simulazione del modello descritto sopra



(b) Simulazione [6]

Come si può vedere dalle immagini, le due simulazioni mostrano comportamenti simili, le prede reagiscono nella stessa maniera alla presenza del predatore; per quanto questa non sia una prova della correttezza effettiva del modello, mostra come, al meno dal punto di vista dei comportamenti, il risultato ottenuto sia paragonabile ai modelli presenti nello stato dell'arte.

8 Analisi di sensitività

Di seguito verranno riassunti i risultati delle prede sopravvissute per diverse configurazione di parametri. Dopo aver fissato il numero di predatori (1, 5, 10), per ognuno di essi, e per ogni combinazione per il numero di prede (100, 300, 600) sono state analizzate diverse configurazione per le aree di ogni agente. Identificheremo con delle lettere le varie configurazioni, di seguito la spiegazione per ognuno di esse.

- **A**: configurazione di default
- **B**: diminuzione aree della preda - preda, diminuzione aree della preda - predatore, aumento area predatore
- **C**: aumento aree della preda - preda, diminuzione aree della preda - predatore, aumento area predatore
- **D**: diminuzione aree della preda - preda, aumento aree preda - predatore, diminuzione area predatore
- **E**: aumento aree della preda - preda, aumento aree della preda - predatore, diminuzione area predatore

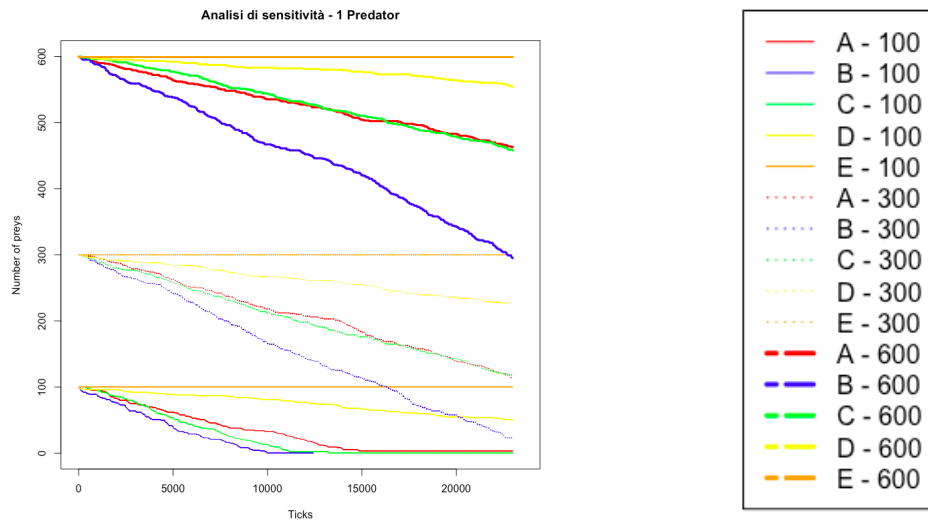
Dal punto di vista numerico i parametri inseriti sono i seguenti:

	A	B	C	D	E
avoidance_area	1	1	1	1	1
parallel_area	8	5	12	5	12
attraction_area	25	15	30	15	30
area_selfish	3	3	3	3	3
area_attention	10	7	7	15	15
visible_area_preay	16	12	12	20	20
capture_area	2	2	2	2	2
attack_area	15	20	20	10	10
visible_area_predator	30	35	35	25	25

8.1 Analisi con 1 predatore

Premettendo che un solo predatore potrebbe impiegare molto più tempo prima di attaccare tutte le prede, si può mostrare come, diminuendo le aree tra le prede (il che significa che le prede più difficilmente troveranno i propri vicini) ma aumentando l'area di attacco del predatore, quest'ultimo riesce meglio ad attaccare (*linea blu*). Si ha una situazione di stallo (*linea arancio*) quando aumentando le aree tra le prede, vengono diminuite le aree del predatore e l'area tra predatore e preda, il che significa che le prede riescono facilmente ad unirsi in banco, il predatore avendo però una visuale molto più bassa non riuscirà ad attaccare.

Una situazione differente si ha quando (*linea gialla*) aumentando la dimensione delle aree preda-predatore, allo stesso momento si ha la diminuzione della dimensione dell'area preda - preda e dell'area del predatore, in questo caso non ci troviamo in una situazione di stallo ma il predatore avrà comunque abbastanza difficoltà a catturare le prede in quanto verrà visto prima dalle prede rispetto a quanto lui faccia con le stesse.



(a) Analisi con 1 Predatore

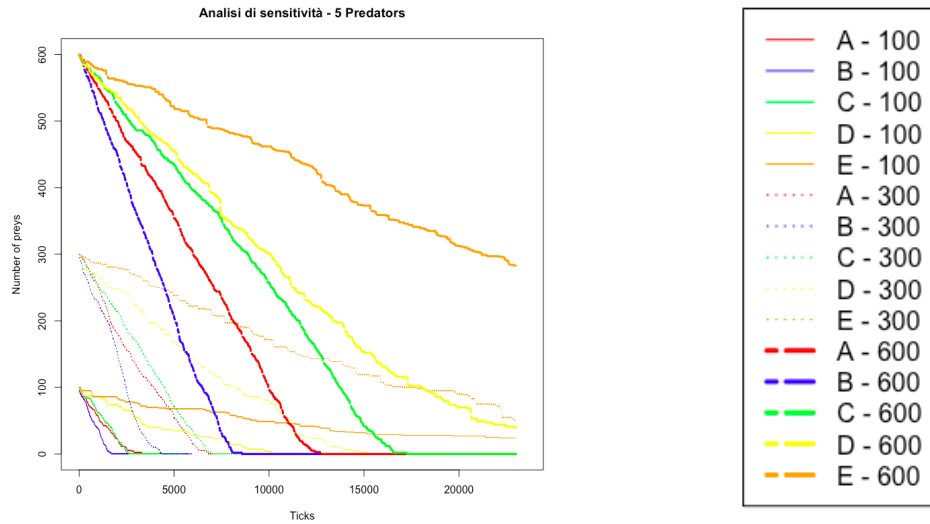
(b) Legenda

8.2 Analisi con 5 predatori

Aumentando a 5 predatori si può già notare un cambiamento rispetto alla configurazione con uno solo. Più predatori sono presenti nell'ambiente più facilmente le prede vengono attaccate. Diminuendo l'area tra le prede, e aumentando l'area del predatore si raggiunge velocemente la situazione in cui non sono presenti prede all'interno dell'ambiente (*linea blu*).

Al contrario del caso precedente vi è un cambiamento all'interno delle dinamiche delle simulazioni al variare delle configurazioni dei parametri; infatti mentre nel caso precedente, a prescindere dal numero di prede, la configurazione di default (*linea rossa*) e la configurazione C (*linea verde*) presentavano poca differenza, in questo caso vi è un cambiamento maggiore.

Come prima, le configurazioni della *linea arancio* e della *linea blu* rispecchiano le simulazioni limite.

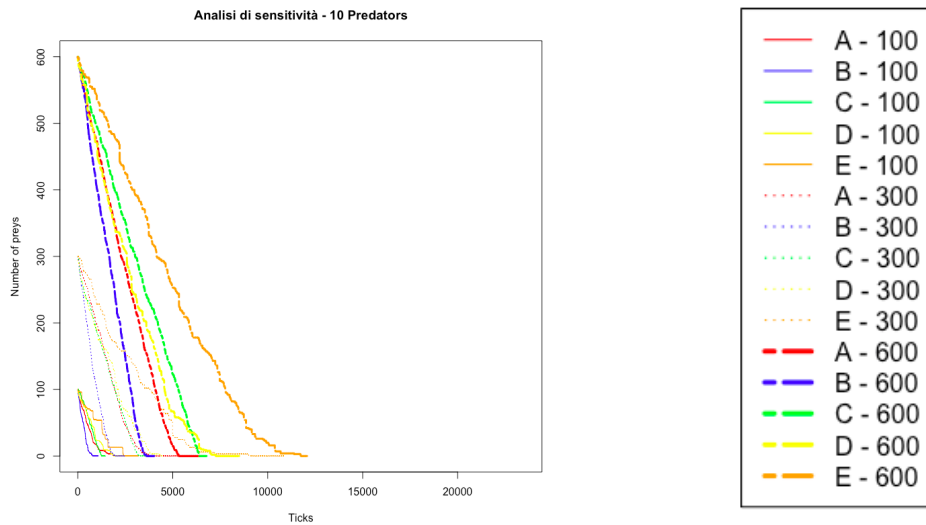


(a) Analisi con 5 Predatori

(b) Legenda

8.3 Analisi con 10 predatori

Le differenze, all'aumentare del numero di predatori, rimangono praticamente invariate. Ciò che ci si aspetterebbe accadere in simulazioni con un singolo predatore avviene in modo accelerato all'aumentare del numero di predatori. L'unica differenza che si può notare tra l'ultimo grafico e quelli precedenti è l'inversione reciproca di posizioni tra la configurazione C (*linea verde*) e la configurazione D (*linea gialla*).



(a) Analisi con 10 Predatori

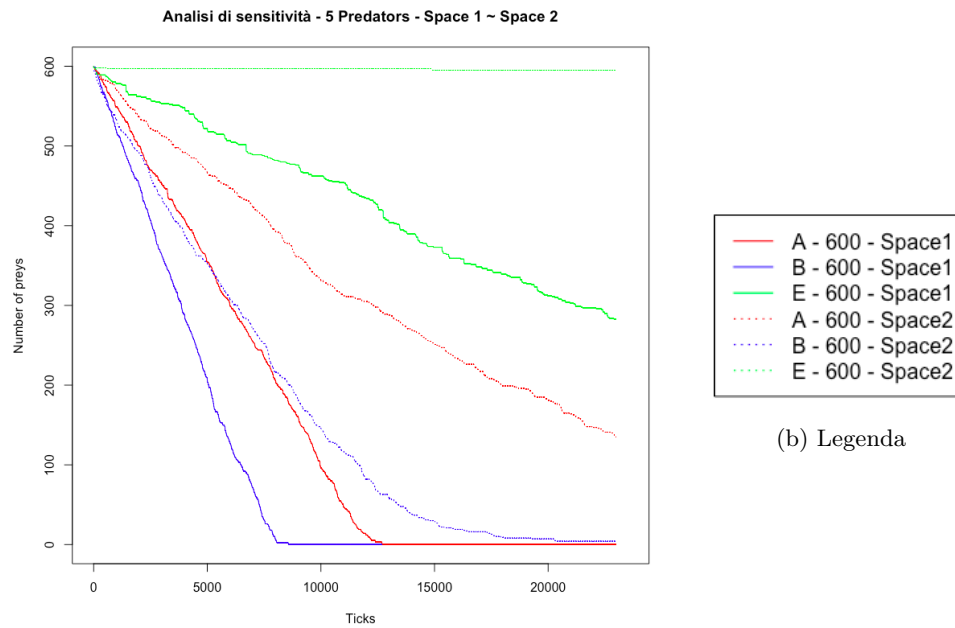
(b) Legenda

8.4 Modifica Dimensione Ambiente

Come ulteriore analisi abbiamo deciso di confrontare alcune configurazioni di parametri tra quelle mostrate in precedenza (A, B ed E), modificando le dimensioni massime dell'ambiente e tenendo fisso il numero di predatori e di prede. Nello specifico, la tabella mostra la variazione delle dimensioni dell'ambiente attuate per effettuare le simulazioni seguenti.

	Spazio 1	Spazio 2
min-pxcor	-70	-200
max-pxcor	70	200
min-pycor	-40	-200
max-pycor	40	200

Spazio 1 identifica la configurazione dell'ambiente utilizzato per le simulazioni i cui risultati sono stati mostrati precedentemente, mentre *Spazio 2* identifica la configurazione di uno spazio più ampio su cui sono state effettuate le simulazioni poi confrontate con i risultati ottenuti in precedenza e mostrati qua sotto.



(a) Analisi con 10 Predatori

Come possiamo notare, anche le dimensioni dell'ambiente sono un parametro fondamentale per le nostre simulazioni; infatti, al variare delle dimensioni impostate, le prede avranno più spazio in cui scappare e saranno quindi meno facilmente catturabili.

8.5 Considerazioni

Escludendo configurazioni di parametri totalmente senza senso rispetto alle dimensioni delle aree di preda e predatore, tutte le simulazioni che abbiamo provato mostrano esattamente il comportamento pensato in fase di modellazione: il singolo predatore non riesce, nel caso in cui attacchi un banco, ad avere successo nell'attacco, in quanto i meccanismi di fuga collettiva sono efficaci (sia in natura che nella nostra implementazione). All'aumentare del numero di predatori essi riescono, in un certo senso collaborando, a catturare le prede non solo quando esse si trovano singolarmente ma approciando il banco da diverse posizioni.

9 Conclusioni

Per sviluppare il progetto abbiamo cercato di utilizzare le conoscenze apprese a lezione come base per la modellazione teorica e matematica della simulazione implementata. Per fare ciò siamo, quindi, partiti dai boids, il modello più completo per la descrizione dell'interazione tra agenti a formare un banco/stormo/gruppo. Partendo da queste regole base abbiamo analizzato diversi papers presenti nello stato dell'arte per capire come fosse già stato affrontato il problema preso in esame; in questo modo siamo riusciti a isolare i meccanismi e le formule utili all'implementazione del nostro progetto. Abbiamo selezionato, tra i molti lavori analizzati, alcuni comportamenti e modelli che fossero il più completi possibile; ovviamente è stato necessario semplificare gli stessi in alcuni punti.

Dopo aver implementato le regole modellate in precedenza, la fase successiva è stata la scelta dei parametri da utilizzare per le nostre simulazioni. Come mostrato nei capitoli precedenti, dopo aver selezionato un set di parametri di default, usati per la maggior parte delle prove effettuate, abbiamo deciso di provare a modificare alcuni parametri (tenendone fissi altri) in modo da poter effettuare analisi di sensitività. In mancanza di una possibilità di validare il modello, in quanto non vi sono risultati di simulazioni di modelli simili al nostro disponibili in rete, la scelta è stata appunto di effettuare analisi per controllare con che modifiche apportate ai parametri, le simulazioni modificassero i propri risultati.

Per concludere, abbiamo cercato di affrontare le varie fasi utili alla simulazione di un sistema complesso e i risultati ottenuti sembrano rispecchiare ciò che ci si sarebbe potuti aspettare da un modello di questo genere.

Il codice del progetto è presente al seguente link :
https://github.com/MonicaMinelli/SchoolingFish_Boids.git

N.B. Partendo dai paper presi in esame, le formule presentate precedentemente sono state adattate al linguaggio utilizzato in fase di implementazione; nello specifico, alcune di esse, presentavano implementazioni simili ma meno costose computazionalmente in Netlogo. Per questo motivo, lo studio teorico presentato nella sezione 4 e nella sezione 5 è stato fondamentale per comprendere le dinamiche di comportamento, ma ha subito alcune variazioni in fase di implementazione.

A Implementazione

A.1 Variabili

Nome variabile	Descrizione
number-of-fishes	Indica il numero di agenti preda da inserire nell'ambiente in fase di creazione del modello.
number-of-predators	Indica il numero di agenti predatore da inserire nell'ambiente in fase di creazione del modello.
avoidance-area	Indica la dimensione dell'area (la misura del raggio) di avoidance delle prede; cioè la distanza alla quale essi cercheranno di allontanarsi dai vicini.
parallel-area	Indica la dimensione dell'area (la misura del raggio) di movimento parallelo e di coesione delle prede; cioè la distanza massima alla quale essi cercheranno di muoversi verso il centro del banco (se esiste già) e in parallelo rispetto ai proprio vicini.
attraction-area	Indica la dimensione dell'area (la misura del raggio) di movimento di attrazione; cioè la distanza massima alla quale essi cercheranno di muoversi verso la preda più vicina.
area-selfish	Indica la dimensione dell'area (la misura del raggio) di movimento in direzione opposta a quella del predatore; cioè la distanza minima alla quale la preda cercherà di muoversi solamente per allontanarsi dal predatore e salvarsi.
area-attention	Indica la dimensione dell'area (la misura del raggio) di movimento della preda in direzione opposta a quella del predatore combinata con il movimento verso il centro del banco; cioè la distanza minima alla quale la preda cercherà di muoversi per allontanarsi dal predatore cercando però di mantenere la direzione anche del banco.
visible-area-prey	Indica la dimensione dell'area (la misura del raggio) di visibilità della preda; cioè la distanza massima alla quale la preda potrà accorgersi della presenza di un predatore.
capture-area	Indica la dimensione dell'area (la misura del raggio) del predatore in cui esso potrà catturare la presa; cioè la distanza massima alla quale la preda sarà mangiata.
attack-area	Indica la dimensione dell'area (la misura del raggio) del predatore in cui esso sferrerà l'attacco; cioè la distanza massima a cui esso potrà effettivamente sferrare l'attacco.
visible-area-predator	Indica la dimensione dell'area (la misura del raggio) del predatore in cui esso potrà iniziare a muoversi verso la preda presente all'interno di essa; cioè la distanza massima alla quale il predatore potrà percepire la presenza delle prede.

A.2 Funzioni

Nome funzione	Descrizione
breed [preys prey]	In questo modo viene creato un insieme di turtles che identificheranno gli agenti prede.
breed [predators predator]	In questo modo viene creato un insieme di turtles che identificheranno gli agenti predatori.
preys-own[...] e predators-own[...]	Questi comandi permettono di assegnare attributi alle differenti classi di agenti create in precedenza.
to setup-patches	Permette di assegnare all'ambiente un colore (e altri parametri).
to setup	Permette di assegnare le istruzioni da svolgere al click del tasto setup nell'interfaccia.
to go	Come sopra, permette di assegnare le istruzioni da svolgere al click del tasto go nell'interfaccia.
to eat	E' la funzione che definisce il comportamento del predatore; essa utilizza altre funzioni internamente che verranno spiegate successivamente.
to find-preysneighbours	Restituisce tutte le altre prede presenti nel raggio di visibilità della preda.
to find-preys-capture-area	Restituisce tutte le prede presenti nell'area di cattura del predatore.
to find-predneighbours	Restituisce gli altri predatori nel raggio di cattura di un predatore in modo da controllare che essi non collidano.
to find-nearest-neighbour -preys	Restituisce la preda più vicina nel set di prede vicine ad un agente.
to find-nearest-pred-pred	Restituisce il predatore più vicino nella lista di predatori vicini ad un agente.
to eat-preysneighbours	E' la funzione che definisce la fase in cui il predatore effettivamente "mangia" la preda, eliminando dall'ambiente l'agente con cui interagisce.
to escape	E' la funzione che definisce in quale modo la preda scapperà dal predatore in base all'area in cui quest'ultimo si troverà rispetto alla preda.
to find-predatorneighbours	Restituisce il set di predatori presenti nell'area di visibilità della preda.
to find-nearest-predators	Restituisce il predatore più vicino nel set di predatori vicini ad una preda.
to cooperative	E' il metodo che definisce il comportamento di cooperative escape.

to selfish-coop	E' il metodo che definisce la direzione in cui fuggire della preda (opposta al predatore) nel caso in cui essa sia in banco.
to selfish	E' il metodo che definisce la fuga in direzione opposta al predatore quando esso è nell'area di selfish della preda.
to movement	E' il metodo che definisce come la preda si muoverà scegliendo ciò che dovrà fare in base alle interazioni con le altre prede.
to find-flockneighbours	Restituisce il set di prede nella parallel area di una determinata preda.
to find-nearest-neighbour	Restituisce la preda più vicina nel set di prede vicine rispetto ad una determinata preda.
to attraction	Definisce il comportamento con il quale la preda si muove in direzione di un'altra preda.
to avoid	Definisce il comportamento per cui una preda, pur cercando di rimanere in banco, non dovrà collidere con un suo vicino.
to avoidPred	E' la funzione che permette ai predatori di non collidere tra di loro.
to parallel	E' la funzione che definisce il movimento parallelo tra prede vicine.
to cohesion	E' il metodo che permette la coesione delle prede in un unico banco.
to turn-away [new-heading max-turn]	Permette di modificare la direzione di movimento nel senso opposto ad un altro agente definendo l'angolo massimo di sterzata.
to turn-towards [new-heading max-turn]	Permette di modificare la direzione di movimento verso un altro agente definendo l'angolo massimo di sterzata.
to turn-at-most [turn max-turn]	Permette di definire se girare a destra o a sinistra in base all'angolo massimo di sterzata fornito.
to-report av-flockneighbours-heading	Restituisce la media delle direzioni di movimento delle prede vicine ad un determinato agente.
to-report av-towards-flockneighbours-heading	Restituisce la media delle direzioni di movimento verso il centro delle prede;fornisce cioè la direzione in cui si trova il centro del banco.

B Listato del codice

```
breed [preys prey]
breed [predators predator]

preys-own[
  nearest-neighbour
  nearest-predator
  predatorneighbours
  flockneighbours
  speed
]

predators-own[
  nearest-preysneighbours
  preysneighbours
  preys-capture-area
  nearest-pred
  predneighbours
  speed
]

to setup-patches
  ask patches [set pcolor blue]
end

to setup
  clear-all
  setup-patches
  create-preys number_of_fishes
  [ set color yellow
    set size 1.2
    set shape "fish"
    setxy random-xcor random-ycor
    set flockneighbours no-turtles]

  create-predators number_of_predators
  [ set color black
    set size 1.8
    set shape "turtle"
    setxy random-xcor random-ycor
    set preysneighbours no-turtles]

  reset-ticks

end

to go
```



```

ask preys [ movement escape ]
ask predators [eat]
repeat 1 [ask preys [fd 0.2] display]
repeat 1 [ask predators [fd 0.2] display]
;; settando fd a valori minori si riduce l'efficienza ma si
    ↪ aumenta
;; la fluidit  dell'animazione
tick
end

to eat
  find-preysneighbours
  find-predneighbours
  ifelse any? predneighbours
    [find-nearest-pred-pred
     avoidPred]
    [ifelse any? preysneighbours
      [find-nearest-neighbour_preys
       ifelse distance nearest-preysneighbours >= attack_area
         [ turn-towards (towards nearest-preysneighbours) 10
           set speed 0.4
           fd speed ]
         [ifelse distance nearest-preysneighbours < attack_area and
           ↪ distance nearest-preysneighbours >= capture_area
            [turn-towards (towards nearest-preysneighbours) 10
              set speed 2.2
              fd speed]
            [eat-preysneighbours]]]
      [turn-towards random 180 30]]
end

to find-preysneighbours
  set preysneighbours other preys in-radius visible_area_predator
end

to find-preys-capture-area
  set preys-capture-area other preys in-radius capture_area
end

to find-predneighbours
  set predneighbours other predators in-radius capture_area
end

to find-nearest-neighbour_preys
  set nearest-preysneighbours min-one-of preysneighbours [
    ↪ distance myself]
end

to find-nearest-pred-pred
  set nearest-pred min-one-of predneighbours [distance myself]

```

```

end

to eat-preysneighbours
  find-preys-capture-area
  if any? preys-capture-area
    [let prey_to_die nearest-preysneighbours
     if prey_to_die != nobody [
       ask prey_to_die [ die ]
     ]
    ]
  end

to escape
  find-flockneighbours
  find-predatorneighbours
  if any? flockneighbours
    [if any? predatorneighbours
     [find-nearest-predators
      ifelse distance nearest-predator > area_selfish and distance
        ↪ nearest-predator <= area_attention
        [cooperative]
        [if distance nearest-predator <= area_selfish
         [selfish]]]]
  end

to find-predatorneighbours
  set predatorneighbours other predators in-radius
  ↪ visible_area_pre
end

to find-nearest-predators
  set nearest-predator min-one-of predatorneighbours [distance
  ↪ myself]
end

to cooperative
  avoid
  parallel
  cohesion
  selfish-coop
end

to selfish-coop
  face nearest-predator
  rt 180
  set speed speed + 0.4
  fd speed
end

to selfish

```

```

    face nearest-predator
    rt 180
    set speed speed + 0.5
    fd speed
end

to movement
    find-flockneighbours
    ifelse any? flockneighbours
    [find-nearest-neighbour
        ifelse distance nearest-neighbour < avoidance_area
        [ avoid
            set speed random-normal 1 0.3
            fd speed ]
        [ifelse distance nearest-neighbour >= parallel_area and
            ↪ distance nearest-neighbour < attraction_area
            [attraction
                set speed random-normal 1 0.3
                fd speed]
            [parallel cohesion
                set speed random-normal 1 0.3
                fd speed]]]
    [turn-towards random 360 45
        set speed random-normal 1 0.3
        fd speed]
end

to find-flockneighbours
    set flockneighbours other preys in-radius parallel_area
end

to find-nearest-neighbour
    set nearest-neighbour min-one-of flockneighbours [distance
        ↪ myself]
end

to attraction
    turn-towards ([heading] of nearest-neighbour) 20
end

to avoid
    find-flockneighbours
    ifelse any? flockneighbours
    [find-nearest-neighbour
        turn-away ([heading] of nearest-neighbour) 5]
    [turn-towards random 360 45
        set speed random-normal 1 0.3
        fd speed]
end

```

```

to avoidPred
  face nearest-pred
  rt 180
end

to parallel
  turn-towards av-flockneighbours-heading 20
end

to cohesion
  turn-towards av-towards-flockneighbours-heading 10
end

to turn-away [new-heading max-turn]
  turn-at-most (subtract-headings heading new-heading) max-turn
end

to turn-towards [new-heading max-turn]
  turn-at-most (subtract-headings new-heading heading) max-turn
end

to turn-at-most [turn max-turn]
  ifelse abs turn > max-turn
  [ ifelse turn > 0
    [ rt max-turn]
    [ lt max-turn]]
  [rt turn]
end

to-report av-flockneighbours-heading
  let x-component sum [dx] of flockneighbours
  let y-component sum [dy] of flockneighbours
  ifelse x-component = 0 and y-component = 0
  [report heading]
  [report atan x-component y-component]
end

to-report av-towards-flockneighbours-heading
  let x-component mean [sin (towards myself + 180)] of
    ↪ flockneighbours
  let y-component mean [cos (towards myself + 180)] of
    ↪ flockneighbours
  ifelse x-component = 0 and y-component = 0
  [report heading]
  [report atan x-component y-component]
end

```

Riferimenti bibliografici

- [1] Nikolai WF Bode, Jolyon J Faria, Daniel W Franks, Jens Krause, and A Jamie Wood. How perceived threat increases synchronization in collectively moving animal groups. *Proceedings of the Royal Society of London B: Biological Sciences*, page rspb20100855, 2010.
- [2] Edward T. Hall. Proxemics. <https://en.wikipedia.org/wiki/Proxemics>.
- [3] James E Herbert-Read, Emil Rosén, Alex Szorkovszky, Christos C Ioannou, Björn Rogell, Andrea Perna, Indar W Ramnarine, Alexander Kotrschal, Niclas Kolm, Jens Krause, et al. How predation shapes the social interaction rules of shoaling fish. *Proc. R. Soc. B*, 284(1861):20171126, 2017.
- [4] Julia K Parrish and Leah Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.
- [5] David JT Sumpter. The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 361(1465):5–22, 2006.
- [6] M Zheng, Y Kashimori, O Hoshino, K Fujita, and T Kambara. Behavior pattern (innate action) of individuals in fish schools generating efficient collective evasion from predation. *Journal of theoretical biology*, 235(2):153–167, 2005.
- [7] Meihong Zheng, Yoshimasa Narita, Yoshiki Kashimori, Osamu Hoshino, and Takeshi Kambara. Effectiveness of allelomimesis of individuals in dynamical response of fish school to emergent affairs. *IPSJ SIGnotes*, 34, 2001.