Target-oriented predator and prey swarm control in obstacle-filled environments

Author: Catalin-Petru Lupau*

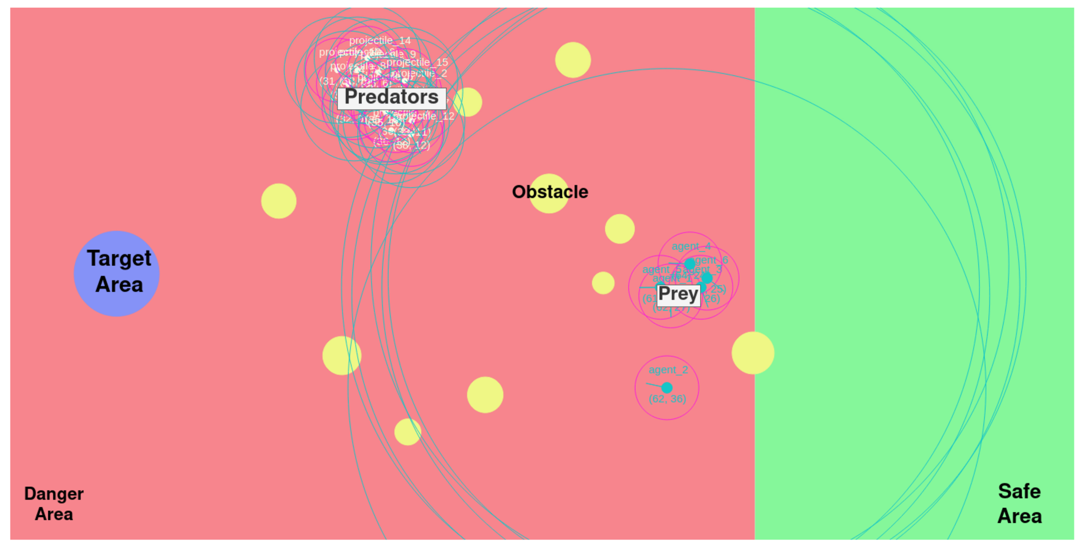
Responsible Professor: Dr. Ranga Rao Venkatesha

Supervisor(s): Ashutosh Simha, Suryansh Sharma

EEMCS faculty Delft University of Technology

*Email: C.P.Lupau@student.tudelft.nl

1. Background



Context

- swarms are groups of interacting entities whose behaviors lead to the emergence of complex behavior that wouldn't be achievable by each part alone
- swarming behavior has evolved independently in many animal species
- understanding swarms can lead to useful applications

Problem under scope

- zero sum game with potentially real-life applications involving a prey swarm and a predator swarm
- the prey swarm tries to reach the target fast and with minimal losses
- the predator swarm tries to destroy the prey agents
- when a predator agent hits a prey agent, both agents get destroyed
- when an agent, predator or prey, hits an obstacle, the agent gets destroyed
- when a predator agent hist the target area, the agent gets destroyed
- prey agents spawn in the safe (green) area
- obstacles spawn randomly in the danger (red) area

2. Research Questions & Method

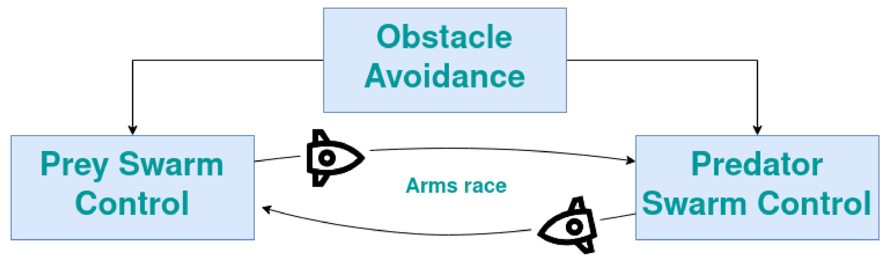
Research Questions

1. Can the number of prey agents that reach the target be increased and their travel time be decreased through the creation or use of smart prey swarm control algorithms? If so, how would such a algorithms work? How would they compare with each other?

2. Given some prey swarm control algorithm, can the number of prey agents that reach the target be decreased or their travel time be increased through the creation or use of smart predator swarm control algorithms? If so, how would such a algorithms work? How would they compare with each other?

Methodology

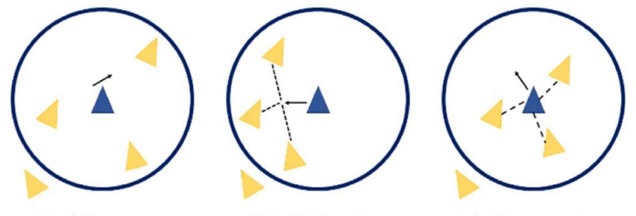
- first examine obstacle avoidance algorithms, since they are needed for both prey and predator.
- develop the prey and predator algorithms iteratively in an arms-race manner.
- benchmark the performance of the algorithms using self-implemented simulation software



3. Algorithms

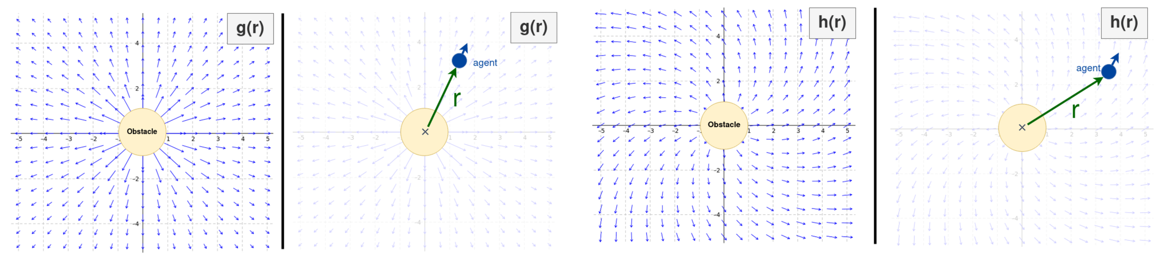
Boids

- computational model emulating flocking (Craig Reynolds)
- behavior: *alignment*, *cohesion*, separation
- easily extendable
- forms the theoretical basis for this paper



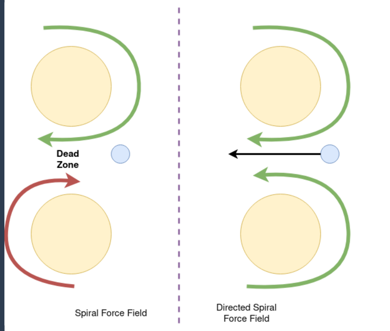
(Tae Jong Choi and Chang Wook Ahn, 2019, see ref. [1])

Obstacle Avoidance



Outwards Field $\vec{g}(\vec{r}) = \frac{w_g}{\|\vec{r}\|^2}\vec{r}$

Spiral Field $\vec{h}(\vec{r}) = (r_x - r_y)\vec{i} + (r_x + r_y)\vec{j} + \frac{w_h}{\|\vec{r}\|^2}\vec{r}$

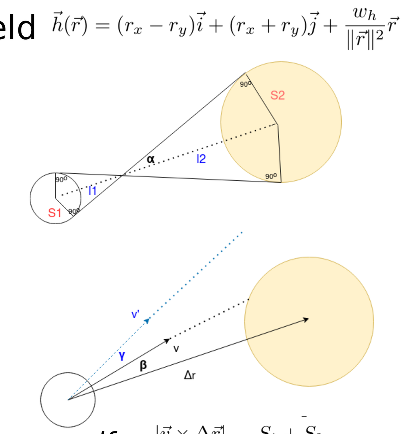


$$\vec{h}_a(\vec{r}) = (r_x - r_y)\vec{i} + (r_x + r_y)\vec{j} + \frac{w_h}{\|\vec{r}\|^2}\vec{r}$$

$$\vec{h}_c(\vec{r}) = (r_x + r_y)\vec{i} + (r_y - r_x)\vec{j} + \frac{w_h}{\|\vec{r}\|^2}\vec{r}$$

$$\vec{h}_{dir}(\vec{r}, \vec{s}) = \begin{cases} \vec{h}_a(\vec{r}), & \|\vec{r}\| \leq d \wedge \vec{s} \cdot \vec{r} < 0 \\ \vec{h}_c(\vec{r}), & \|\vec{r}\| \leq d \wedge \vec{s} \cdot \vec{r} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Directed Spiral Field



if $\frac{|\vec{v} \times \Delta\vec{r}|}{|\vec{v} \cdot \Delta\vec{r}|} \leq \frac{S_1 + S_2}{l_1 + l_2}$

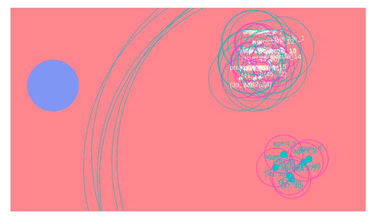
then $\vec{v} := \begin{cases} R(\gamma)\vec{v}, & \text{if } \vec{v} \cdot \Delta\vec{r} \geq 0 \\ R(-\gamma)\vec{v}, & \text{otherwise} \end{cases}$

Steer away

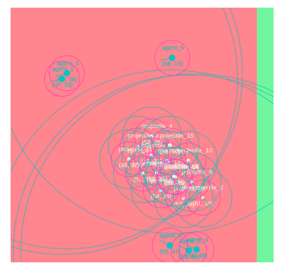
Prey Swarm Control

$$\vec{j}_i = \begin{cases} \text{random\_dir}(), & \text{if} \|\vec{p_k} - \vec{p_i}\| \leq p\_d \text{ for some } k \in P \\ 0, & \text{otherwise} \end{cases}$$

Jump



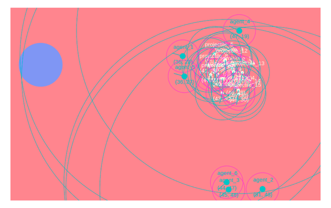
Evasive $\vec{e_i} = \sum_{k \in N_p} (\vec{p_i} - \vec{p_k})$



$$\vec{ca}_i = \frac{1}{|A_i|} \sum_{j \in A_i} \vec{p}_j$$

$$\vec{a}_i = \vec{p}_i - \vec{ca}_i$$

Split

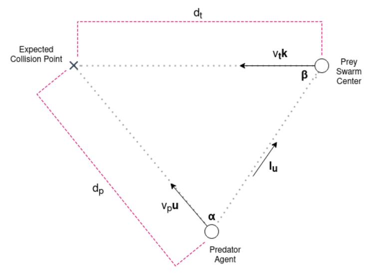


Explode $\vec{X}_i = \frac{\vec{p_i} - \vec{c}}{\|\vec{p_i} - \vec{c}\|^2}$

Predator Swarm Control

$$\vec{f}_i = S\vec{s}_i + K\vec{k}_i + M\vec{m}_i + \text{C.A.} + T\frac{1}{|B|}(\sum_{j \in B} \vec{p}_j - \vec{p}_i)$$

Center Steering



$$\vec{v}_t = (\frac{1}{|B|}\sum_{i \in B} \vec{v}_i)\frac{\vec{t} - \frac{1}{|B|}\sum_{i \in B}\vec{p}_i}{\|\vec{t} - \frac{1}{|B|}\sum_{i \in B}\vec{p}_i\|}$$

$$v_t = \|\vec{v}_t\|, \vec{k} = \frac{\vec{v}_t}{\|\vec{v}_t\|}$$

$$c_u = \frac{v_t}{v_p}\|\vec{k} \times \vec{l}_u\|$$

$$\begin{matrix} \sqrt{1 - c_u^2} & -c_u \\ c_u & \sqrt{1 - c_u^2} \end{matrix} \Big] \vec{l}_u$$

Proportional Navigation

- make use of a clustering algorithm to identify sub-swarms
- divide the predator swarm accordingly, such that each predator sub-swarm deals with one prey sub-swarm
- K-Means could be used for this purpose, however doing it every frame is computationally expensive.

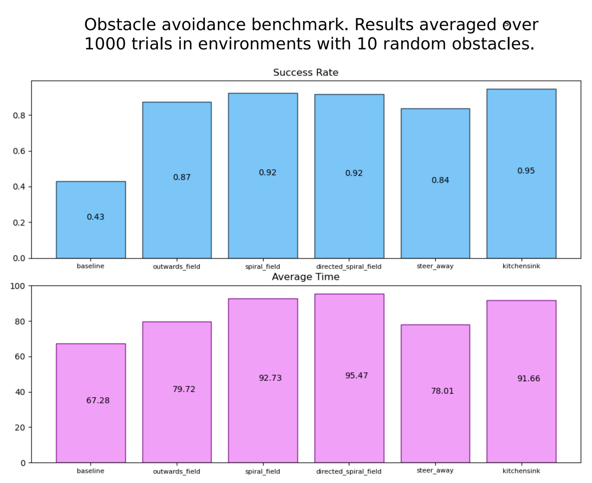
Clustering

4. Results

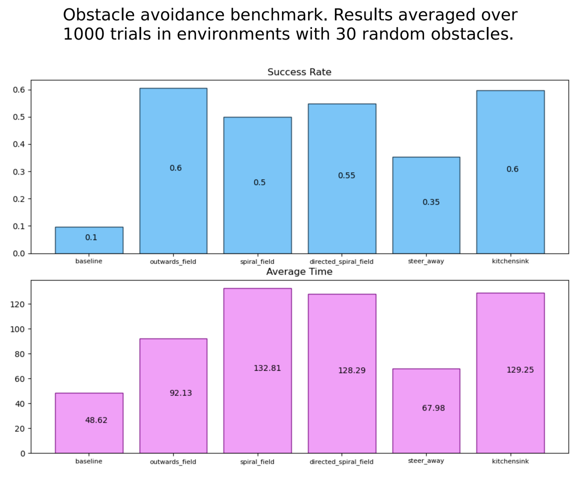
All algorithms described in the previous section were benchmarked using the simulation framework created for this research. During the benchmarking process the success rate (average percentage of prey agents succeeding in reaching the target) and the average trial time (how long it takes on average for all prey agents to reach the target) were measured.

Two types of experiments were performed, *obstacle avoidance* and *predator vs. prey*. The *obstacle avoidance* experiment aimed to test the capabilities of the collision avoidance strategies common to both the prey and predator control algorithms. In contrast, the *predator vs. prey* experiment was done to analyze how the 2 categories of algorithms behave against each other.

The outcomes of the experiments were to a high extent influenced by the chosen hyperparameters.



Algorithm	Success Rate	Avg. Time [s]
Baseline	0.43	67.28
Outwards Field	0.87	79.72
Spiral Field	0.92	92.73
Directed Spiral Field	0.92	95.47
Steer Away	0.84	78.01
Kitchensink	0.95	91.66



Algorithm	Success Rate	Avg. Time [s]
Baseline	0.1	48.62
Outwards Field	0.6	92.13
Spiral Field	0.5	132.81
Directed Spiral Field	0.55	128.29
Steer Away	0.35	67.98
Kitchensink	0.6	129.25

Predator vs. prey

Prey control algorithms benchmarked against center steering predators. Results averaged over 1000 trials in environments without obstacles.

Algorithm	Success Rate	Avg. Time [s]
Baseline	0.54	71.63
Evasive	0.89	93.98
Explode	0.65	89.99
Explode-Evasive	0.88	121.42
Jump	0.52	76.23
Split	0.62	111.62
Split-Evasive	0.94	137.47

Prey control algorithms benchmarked against center steering predators. Results averaged over 1000 trials in environments with 10 random obstacles.

Algorithm	Success Rate	Avg. Time [s]
Baseline	0.45	87.77
Evasive	0.57	108.22
Explode	0.63	113.79
Explode-Evasive	0.83	133.39
Jump	0.45	95.38
Split	0.63	133.9
Split-Evasive	0.81	152.76

Prey control algorithms benchmarked against png predators. Results averaged over 1000 trials in environments without obstacles.

Algorithm	Success Rate	Avg. Time [s]
Baseline	0.13	61.24
Evasive	0.22	134.57
Explode	0.79	90.52
Explode-Evasive	0.66	162.25
Jump	0.13	65.08
Split	0.64	112.64
Split-Evasive	0.53	195.33

Prey control algorithms benchmarked against png predators. Results averaged over 1000 trials in environments with 10 random obstacles.

Algorithm	Success Rate	Avg. Time [s]
Baseline	0.37	92.78
Evasive	0.34	151.37
Explode	0.58	118.32
Explode-Evasive	0.5	195.2
Jump	0.35	100.88
Split	0.51	138.3
Split-Evasive	0.46	200.38

5. Conclusion

- most proposed algorithm perform better than the baseline (no algorithm at all), which means they are all successful at least to some degree at their intended task.
- the spiral field obstacle avoidance algorithms performed worse than expected. A formal analysis of possible deadlock situations could lead to potential improvements to the algorithms.
- the cluster predator was too computationally expensive for benchmarking. A fast algorithm for tackling prey sub-swarms would be a good research direction.
- neural network approaches to both prey and predator control could be looked into in a future research
- the jump predator avoidance strategy was not a good idea, since it is as bad as the baseline

Notation Legend & Parameters

Notation Legend

- b_i - boid i
- $\vec{p}_i$ - position vector of boid i
- $\vec{v}_i$ - velocity vector of boid i
- $\vec{N}_i$ - neighbors of boid i
- $\vec{c_i}$ - mean position of neighbors
- $\vec{s}_i$ - separation force of boid i
- $\vec{k}_i$ - cohesion force of boid i
- $\vec{m}_i$ - matching force of boid i
- $\vec{f}_i$ - steering force of boid i
- $S, K, M, A, T, E, G, H, J, X$ - real coefficients
- $\vec{g}(\vec{r})$ - outwards force field
- w_g - outwards field strength
- d - threshold distance in various contexts
- $\vec{h}(\vec{r})$ - spiral force field
- w_h - spiral force field outwards component strength
- $\vec{h}_a(\vec{r})$ - anti-clockwise spiral force field
- $\vec{h}_c(\vec{r})$ - clockwise spiral force field
- $\vec{h}_{dir}(\vec{r}, \vec{s})$ - directional force field
- $\vec{s}$ - steering direction (difference between the target's position vector and the boid's position vector)
- $\vec{t}$ - position vector of target area
- $\vec{o_j}$ - position vector of object j
- S_1 - the radius of the agent (see diagram)
- S_2 - the radius of the obstacle (see diagram)
- l_1 - distance between agent center and boundary intersection point (see diagram)
- l_2 - distance between obstacle center and boundary intersection point (see diagram)
- α - boundary angle (steer away context)
- β - actual steering angle with respect to obstacle (i.e. if β is within $+\alpha$ and $-\alpha$, the agent will hit the target (steer away context)
- $\Delta\vec{r}$ - relative position vector between agent and obstacle
- $R(x)$ - 2D rotational matrix
- γ - small steer away adjustment angle
- A_i - set of anti-neighbors of agent i
- C.A. - genetic collision avoidance term.
- $\vec{f_i}$ - steering force driving the agents.
- B - set of all prey boids.
- P - set of all predators.
- $\vec{e}_i$ - evasive force
- $\vec{ca}_i$ - mean position of neighbors of agent i that are part of a different sub-swarm.
- $\vec{a}_i$ - anti-neighbor force
- d_t - distance between target and expected collision point (in the context of the collision pyramid).
- d_p - distance between predator and expected collision point (in the context of the collision pyramid).
- v_p - predator / pursuer speed
- v_t - approximated target / prey speed

Hyperparameter values

- M - 0.05
- K - 0.005
- S - 0.05
- w_g - 10
- G - 1.0
- H - 1.0
- T - 0.1
- w_h = 10
- E - 0.01
- X - 20.0
- explosion time - 50
- J - 3.0
- A - 200.0
- anti neighbor time - 30.0
- anti neighbor distance - 50.0
- perception distance - 30
- swarm distance - 3
- number of prey agents - 6
- number of predator agents - 15

References

[1] Tae Jong Choi, Chang Wook Ahn, Artificial life based on boids model and evolutionary chaotic neural networks for creating artworks, Swarm and Evolutionary Computation, Volume 47, 2019, Pages 80-88, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2017.09.003, fig. 5