Decentralized methods for Weapon-Target Assignment

Implementation of [1] by Landon Shumway

EE 682R – Multi-agent Systems

4/15/24

[1] K. Volle, J. Rogers, and K. Brink, “Decentralized Cooperative Control Methods for the Modified Weapon–Target Assignment Problem,” *Journal of Guidance, Control, and Dynamics*, Jul. 2016, doi: 10.2514/1.G001752.

BYU

Outline

- Motivation of Problem
- Background Literature
- Set up
- Methods
- Results
- Conclusion/Future Work/Questions

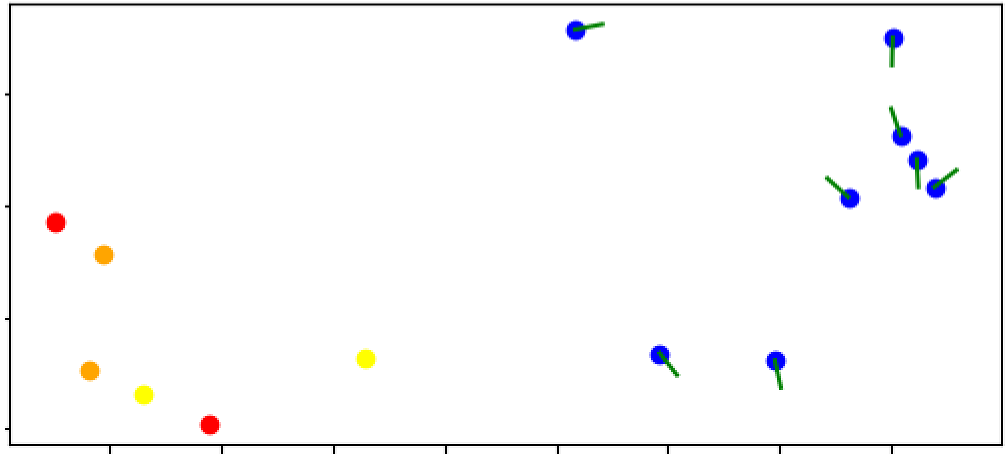
Motivation of Problem

- Weapon-Target Assignment Problem
  - Set of mobile agents/weapons
  - Set of stationary targets
  - Find the optimal weapon-target pairing *Q:*

$$Q = \{(i, j) | i \in I, j \in J\}$$

$$I = \{1, \ldots, N\}$$

$$J \subseteq \{1, \ldots, M\}$$



BYU

Background Literature

Paper implementation:

- K. Volle, J. Rogers, and K. Brink, “Decentralized Cooperative Control Methods for the Modified Weapon–Target Assignment Problem,” Journal of Guidance, Control, and Dynamics, Jul. 2016, doi: 10.2514/1.G001752.

WTA first explored (linear approximation of nonlinear aspects):

- A. S. Manne, “A Target-Assignment Problem,” Operations Research, vol. 6, no. 3, pp. 346–351, Jun. 1958, doi: 10.1287/opre.6.3.346.

Integer programming problem:

- R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, “Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem,” Operations Research, vol. 55, no. 6, pp. 1136–1146, Dec. 2007, doi: 10.1287/opre.1070.0440.

Game Theory:

- G. Arslan, J. R. Marden, and J. S. Shamma, “Autonomous Vehicle-Target Assignment: A Game-Theoretical Formulation,” Journal of Dynamic Systems, Measurement, and Control, vol. 129, no. 5, pp. 584–596, Apr. 2007, doi: 10.1115/1.2766722.
- A. Chapman, R. A. Micillo, R. Kota, and N. Jennings, “Decentralised Dynamic Task Allocation: A Practical Game–Theoretic Approach,” presented at the The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’09) (10/05/09 - 15/05/09), May 2009, pp. 915–922. Accessed: Apr. 10, 2024. [Online]. Available: https://eprints.soton.ac.uk/267066/

Simulated Annealing:

- E. E. Witte, R. D. Chamberlain, and M. A. Franklin, “Task assignment by parallel simulated annealing,” in Proceedings., 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors, Sep. 1990, pp. 74–77. doi: 10.1109/ICCD.1990.130165.
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” Science, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

Hybrid Solutions:

- Z.-J. Lee and W.-L. Lee, “A Hybrid Search Algorithm of Ant Colony Optimization and Genetic Algorithm Applied to Weapon-Target Assignment Problems,” in Intelligent Data Engineering and Automated Learning, J. Liu, Y. Cheung, and H. Yin, Eds., Berlin, Heidelberg: Springer, 2003, pp. 278–285. doi: 10.1007/978-3-540-45080-1-37.
- M. Alighanbari and J. P. How, “Decentralized Task Assignment for Unmanned Aerial Vehicles,” in Proceedings of the 44th IEEE Conference on Decision and Control, Dec. 2005, pp. 5668–5673. doi: 10.1109/CDC.2005.1583066.
- O. Shehory and S. Kraus, “Methods for task allocation via agent coalition formation,” Artificial Intelligence, vol. 101, no. 1, pp. 165–200, May 1998, doi: 10.1016/S0004-3702(98)00045-9.

Newer Methods:

- J. Guo, G. Hu, Z. Guo, and M. Zhou, “Evaluation Model, Intelligent Assignment, and Cooperative Interception in Multimissile and Multitarget Engagement,” IEEE Transactions on Aerospace and Electronic Systems, vol. 58, no. 4, pp. 3104–3115, Aug. 2022, doi: 10.1109/TAES.2022.3144111.
- B. Gaudet, K. Drozd, and R. Furfaro, “Deep Reinforcement Learning for Weapons to Targets Assignment in a Hypersonic strike.” arXiv, Oct. 27, 2023. doi: 10.48550/arXiv.2310.18509.

Novel Contribution

- When defined as an optimization problem, there are many ways to define what "optimal" means
  - Most destroyed targets, most destroyed high-priority targets, etc.
- This paper proposes three new cost functions (beyond the traditional cost function), that induce different behaviors depending on the goal of the scenario

Set up – Agent Model

- 2DOF (constant velocity and altitude)
  - Commanded heading:

$$\psi_{i,com} = \arctan\left(\frac{e_j - e_i}{n_j - n_i}\right)$$

  - Control law:

$$\dot{\psi} = k_\psi(\psi_{i,com} - \psi_i)$$

  - Determining if a target is within range:

$$\dot{d}_i = \frac{v_h}{\sqrt{(n_i - n_j)^2 + (e_i - e_j)^2}} d_i \qquad \frac{v_h}{\dot{d}_i} \leq \gamma_{\max}$$

BYU

Set up – Target Kill Probability

- Each target has a desired kill probability:

$$Pk_{\text{des},j} \ \text{(where } 0 \leq Pk_{\text{des},j} \leq 1)$$

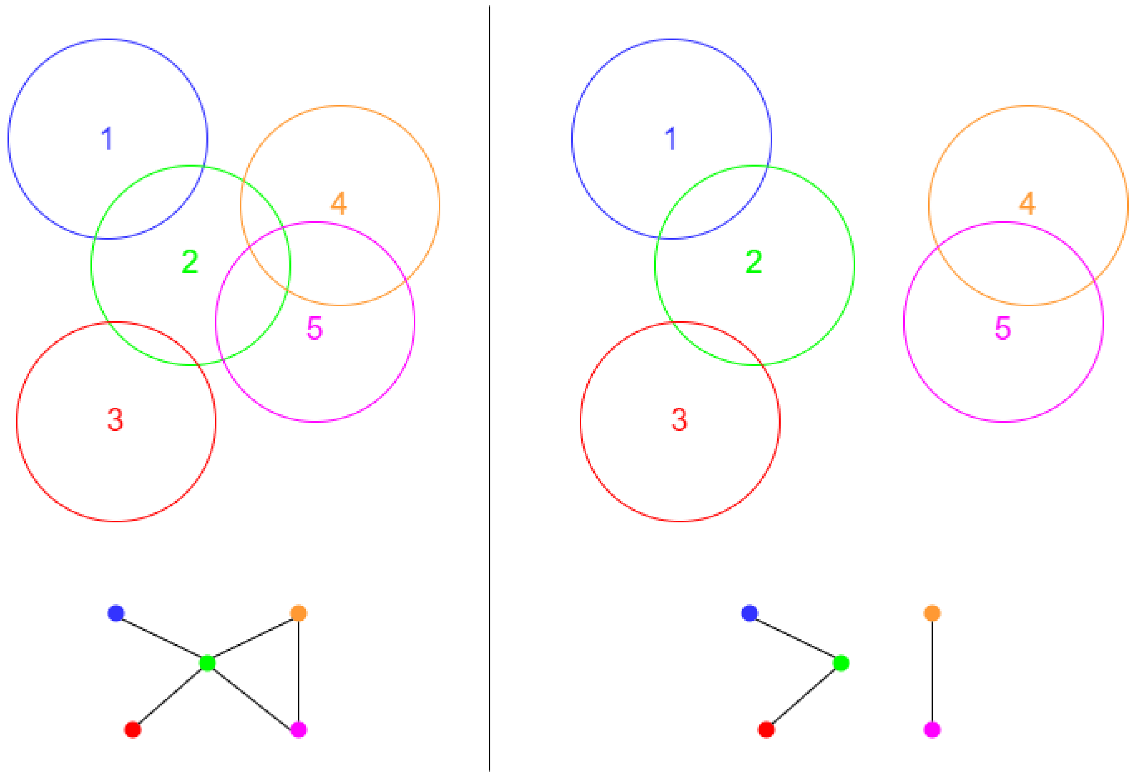
- Calculated kill probability:

$$Pk_{\Sigma,j} = 1 - \prod_{i=1, q_i=j}^{N} \underbrace{(1 - \overbrace{Pk_{i,j}}^{\text{Weapon effectiveness of agent } i} + Pk_{i,j}\overbrace{Pa_{i,j}}^{\text{Probability that agent } i \text{ gets attrited}})}_{\text{Probability of agent } i \text{ destroying target } j}$$

BYU

Set up - Communication

- Agents select targets in a turn-based auction algorithm and broadcast their decision to all agents within its communication range
- This decision gets re-broadcast in a daisy-chain network
- Communication is assumed to occur much faster than decision-making rounds, which are assumed to be on a much shorter timescale than the arrival time from agents to targets
- Each agent maintains a current “estimate” with the most up-to-date information it has on weapon-target pairs. It uses this estimate when deciding which target to pursue
- Broken connections in the network result in out-of-date estimates
    - Not explicitly accounted for in the methods of this paper, but recognized



BYU

Set up - Optimization

- Greedy search
- Using its estimate of the current assignment set Q, each agent i selects the target j that minimizes some cost function the most:

$$q_i = \arg\min_{j \in J} C(Q_j)$$

  - Q_j is the resulting global assignment set if weapon i selects target j
- Greedy search does not always find the optimal solution, but consistently finds a local minimum
- Each agent uses its own estimate and executes its own greedy search, making this method fully decentralized

Set up – Traditional Cost Function

- Cost function is the sum of the the probability that each target is *not* destroyed multiplied by its desired kill probability ("value" term)

$$C_{\mathrm{T}}(Q) = \sum_{j=1}^{M} (1 - Pk_{\Sigma,j}) Pk_{\mathrm{des},j}$$

- The higher the calculated kill probability is, the smaller the resulting cost is
- The desired kill probability weights the importance of each target
- Problem: when there is a large disparity in target values, there is nothing stopping weapons from over-assigning a high-priority target even when its desired kill probability has already been reached
    - Waste of weapon resources

BYU

Methods – Sufficiency Threshold CF

- Provides no incentive to engage a target whose desired kill probability has already been met

$$C_{\mathrm{ST}}(Q) = \sum_{j=1}^{M} \begin{cases} 0 & Pk_{\Sigma,j} > Pk_{\mathrm{des},j} \\ \frac{Pk_{\mathrm{des},j} - Pk_{\Sigma,j}}{(1 - Pk_{\mathrm{des},j})^{\alpha}} & Pk_{\Sigma,j} \leq Pk_{\mathrm{des},j} \end{cases}$$

- Denominator term sets the target "value" as a nonlinear function of $Pk_{des,j}$
  - Alpha is a tuning parameter (higher alpha = more weight to $Pk_{des,j}$)

Methods – Enforced Tiering CF

- Divide the targets into τ tiers (typical to group by $Pk_{des,j}$ values):

  $$T \in 1, \ldots, \tau$$

  Lower value for *T* denotes a higher priority tier

- Cost function sums over tiers:

$$C_{\mathrm{ET}}(Q) = \sum_{T=1}^{\tau} \begin{cases} C_{\mathrm{ST}}(Q_T) & (Pk_{\Sigma,j} \geq Pk_{\mathrm{des},j} \forall j \in t, \forall t < T) \\ P_T & \text{else} \end{cases}$$

- In regular terms: if a higher tier contains targets whose $Pk_{des,j}$ have not been met, enforce some penalty P_T. Else, compute the sufficiency threshold cost function for targets within the current tier
- Penalty must be higher than $C_{ST,max}$

BYU

Methods – Completion CF

- Encourages reaching the $Pk_{des,j}$ for as many targets as possible:

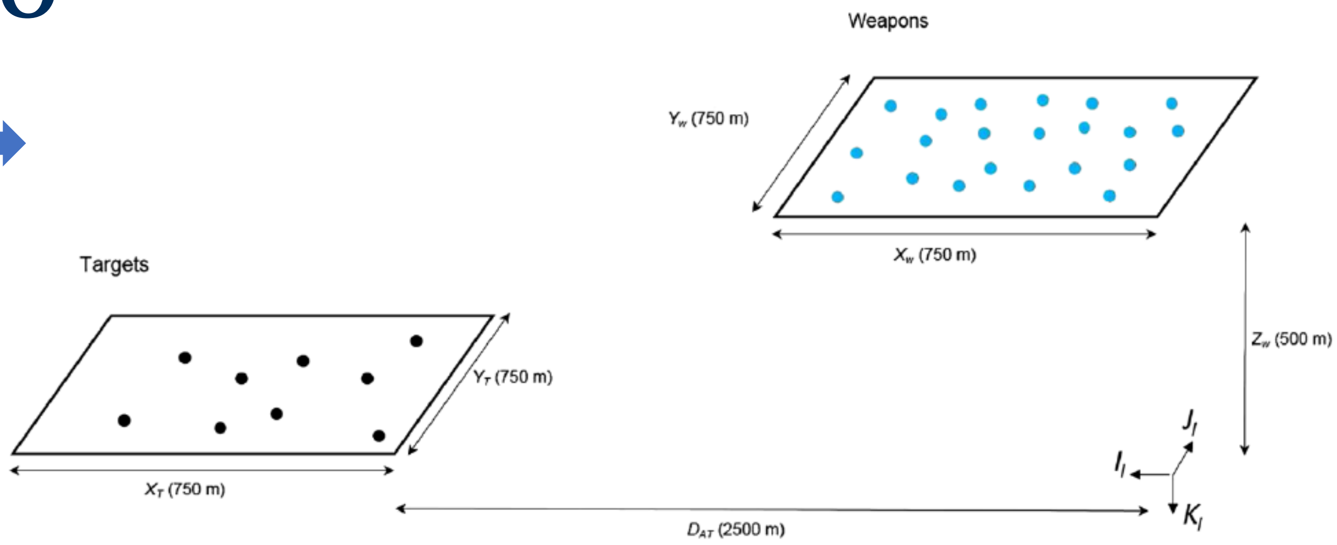
$$C_{\mathrm{C},i}(Q_j) = \frac{\ln(1 - Pk_{\mathrm{des},j}) - \ln(1 - Pk_{\Sigma,j})}{\ln(1 - Pk_{i,j} + Pk_{i,j}Pa_{i,j})}$$

- Results in targets with lower $Pk_{des,j}$ being targeted first
- Derivation is too complex to cover in this presentation, but the basic idea is that agent *i* calculates how many additional weapons of its own effectiveness would be needed to reach $Pk_{des,j}$ for target *j*. This calculation is done for all targets, and the target that requires the least amount of additional weapons is selected.

BYU

Results – Scenario

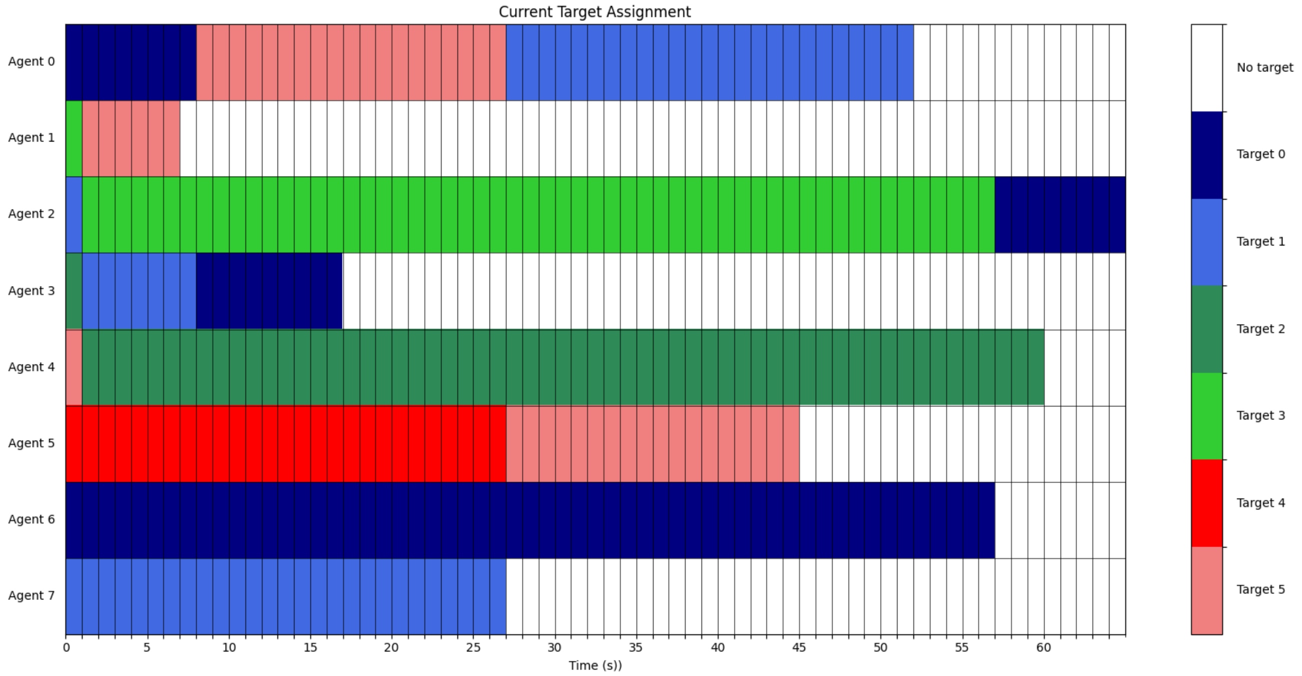
- Engagement geometry
- 6 Targets
  - Targets 0, 1: $Pk_{des,j} = 0.9$
  - Targets 2, 3: $Pk_{des,j} = 0.8$
  - Targets 4, 5: $Pk_{des,j} = 0.7$



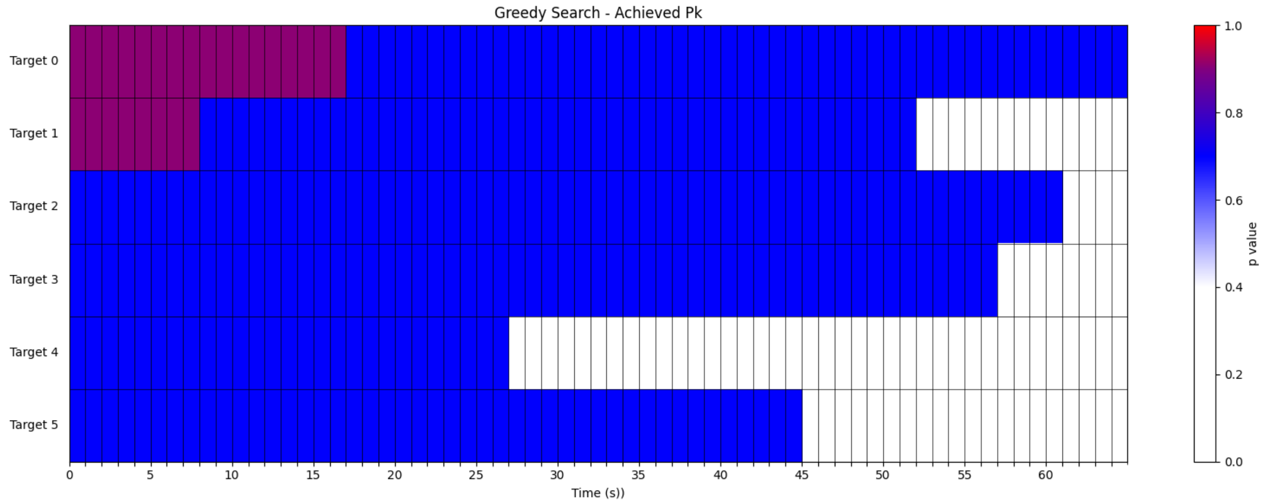
- 8 Weapons
  - All have weapon effectiveness of 0.7
    - Upon collision, the agent is destroyed and the algorithm randomly decides if the target is destroyed with a probability equal to the weapon effectiveness of the agent
    - 0.7 is quite low (0.9 is more accurate/reasonable), but having targets fail to be destroyed induces switching agent/target assignments more, which better illustrates the differences between cost functions

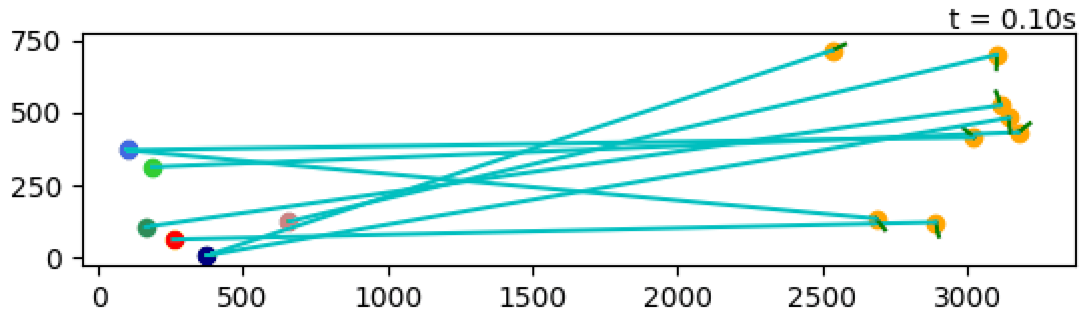
BYU

Results – Traditional CF

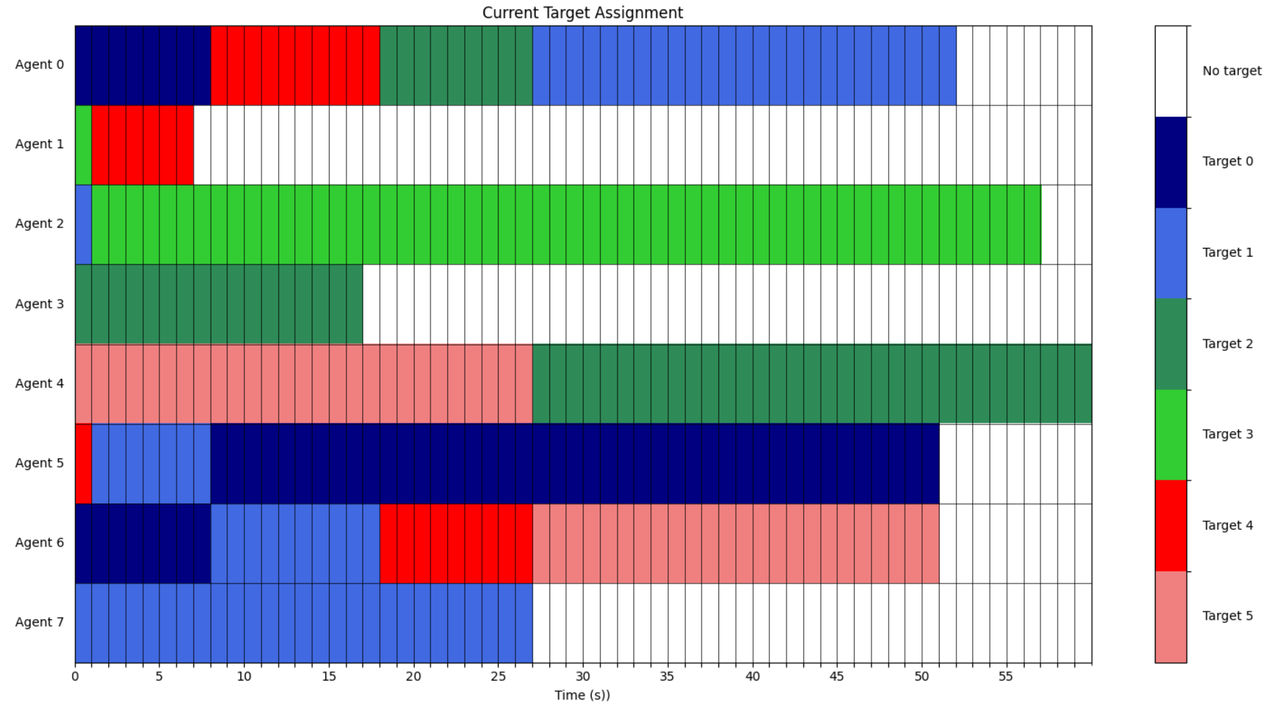


Time (s)	Event
7.1	Agent 1 gets attrited
17.1	Agent 3 gets attrited
26.9	Agent 7 gets attrited
45.4	Agent 5 collides with Target 5; Target 5 is destroyed
52.0	Agent 0 collides with Target 1; Target 1 is destroyed
56.6	Agent 6 collides with Target 0; Target 0 survives
60.3	Agent 4 collides with Target 2; Target 2 survives
64.6	Agent 2 collides with Target 0; Target 0 survives

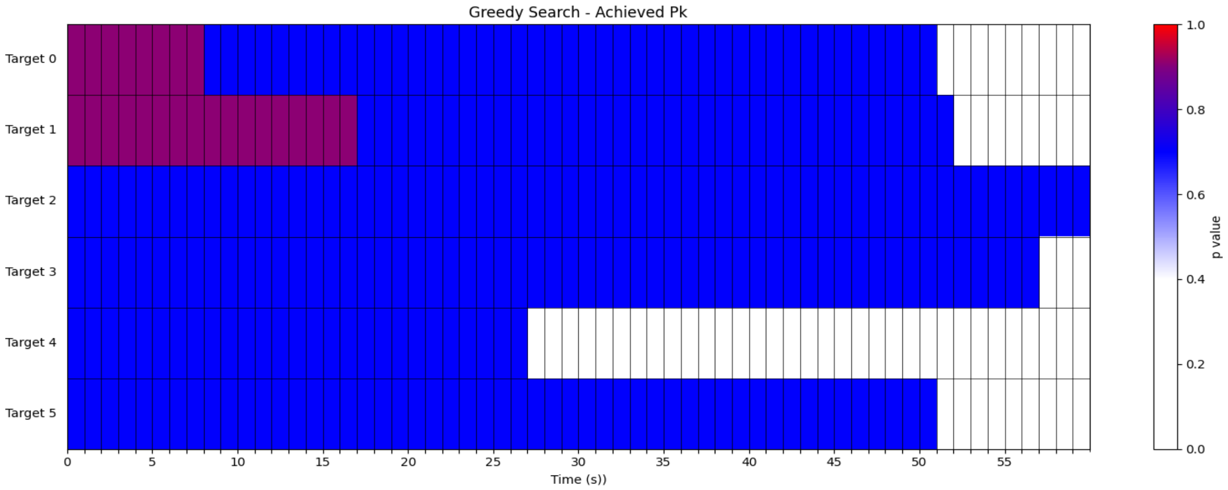


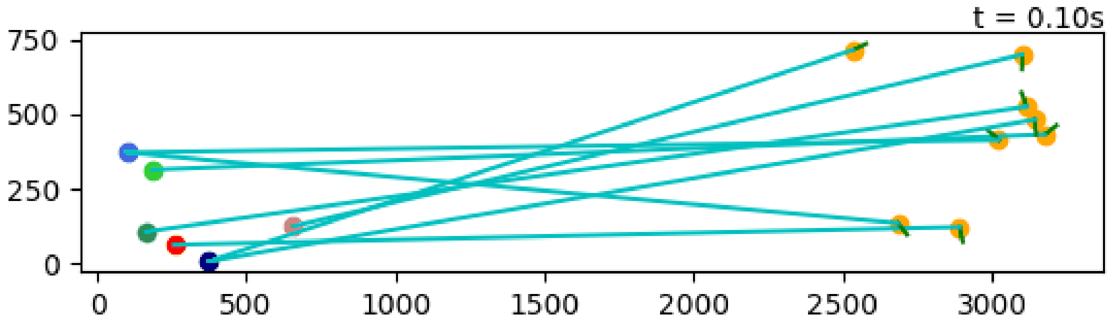


Results – Sufficiency Threshold CF

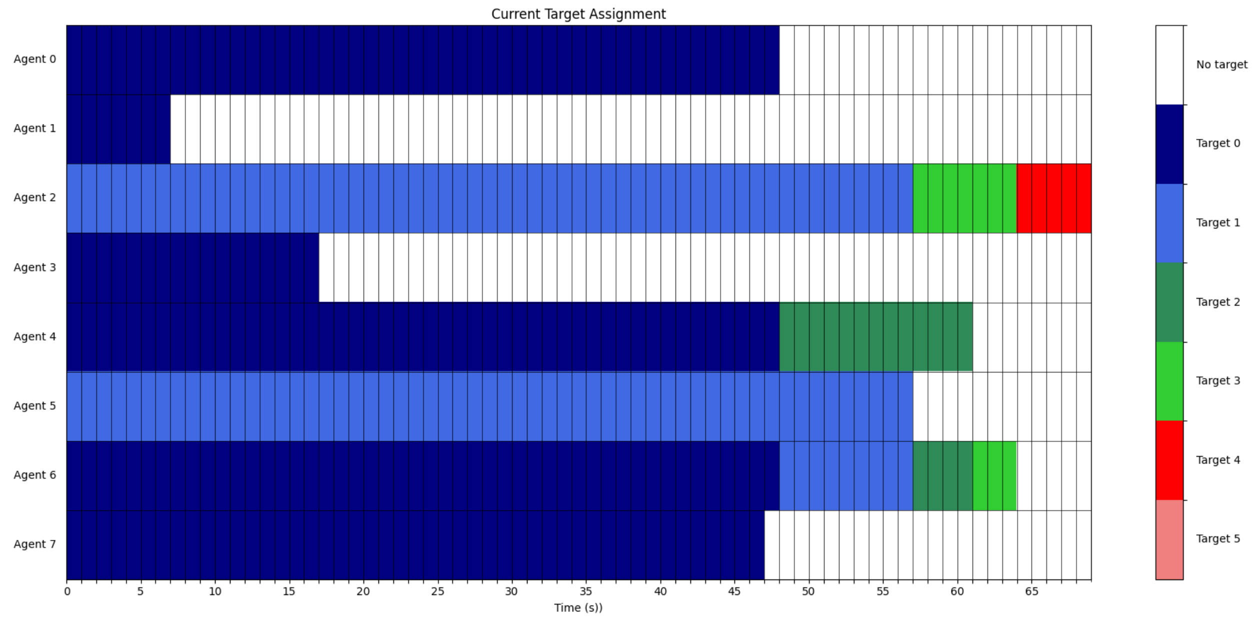


Time (s)	Event
7.1	Agent 1 gets attrited
17.1	Agent 3 gets attrited
26.9	Agent 7 gets attrited
50.8	Agent 6 collides with Target 5; Target 5 survives
51.1	Agent 5 collides with Target 0; Target 0 is destroyed
51.6	Agent 0 collides with Target 1; Target 1 is destroyed
56.9	Agent 2 collides with Target 3; Target 3 survives
63.0	Agent 4 collides with Target 2; Target 2 survives

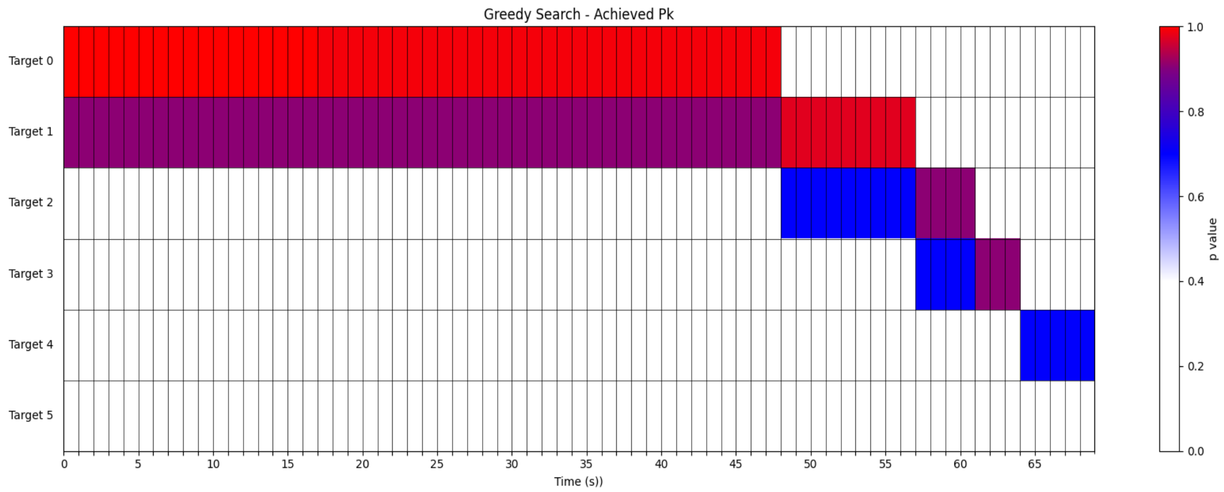


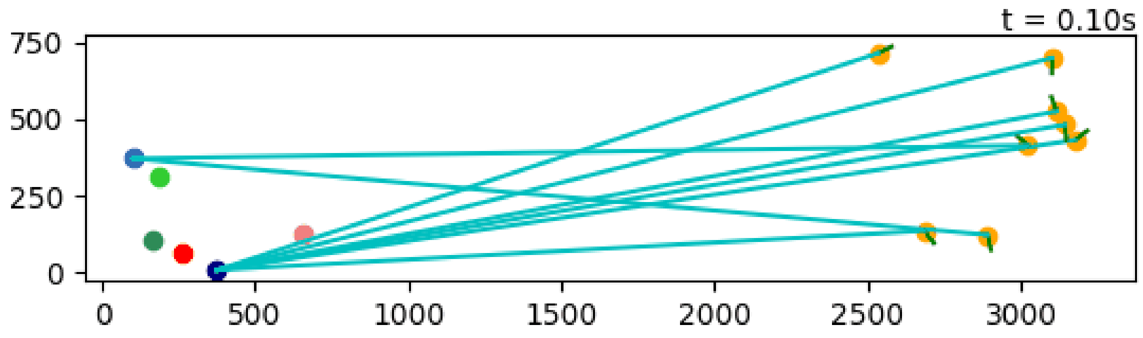


Results – Enforced Tiering CF

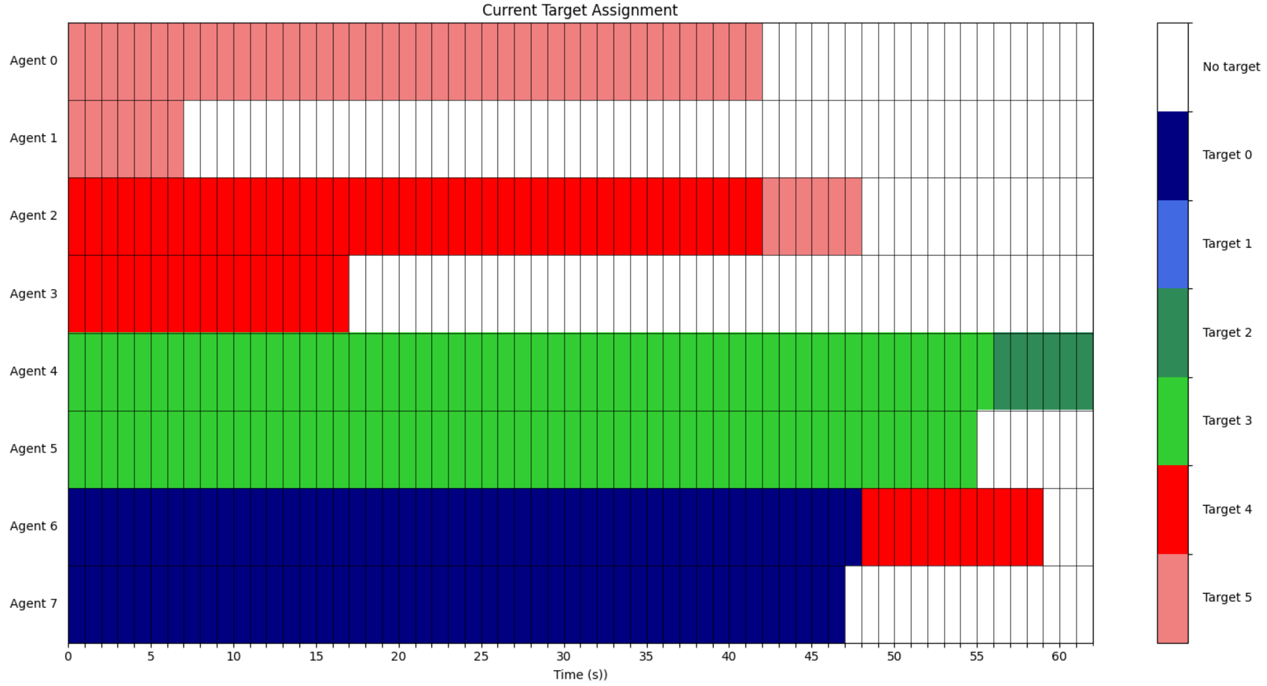


Time (s)	Event
7.1	Agent 1 gets attrited
17.0	Agent 3 gets attrited
47.4	Agent 7 collides with Target 0; Target 0 survives
47.8	Agent 0 collides with Target 0; Target 0 is destroyed
56.9	Agent 5 collides with Target 1; Target 1 is destroyed
60.6	Agent 4 collides with Target 2; Target 2 is destroyed
63.9	Agent 6 collides with Target 3; Target 3 is destroyed
69.0	Agent 2 collides with Target 4; Target 4 survives

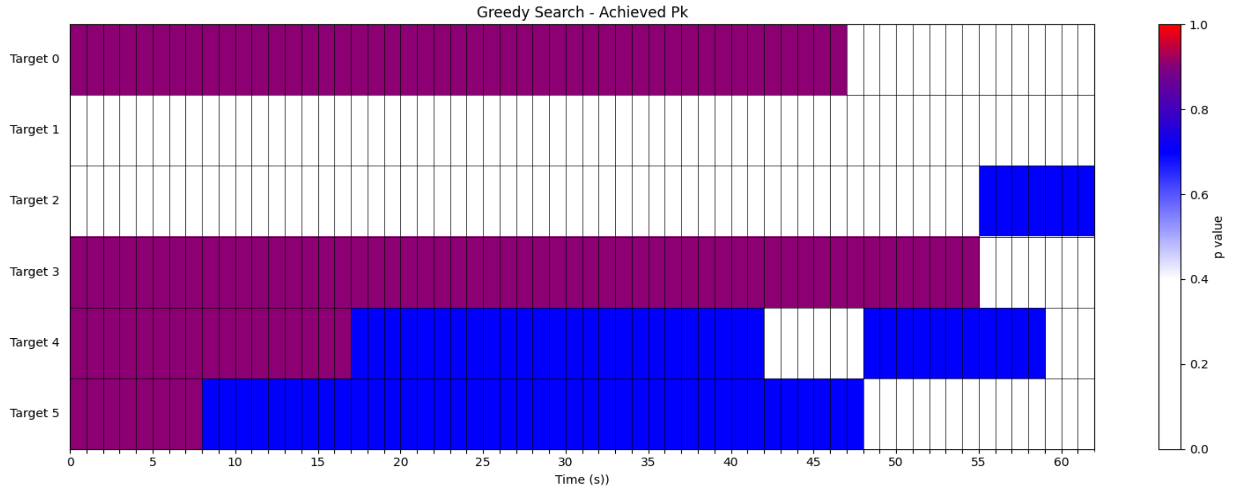


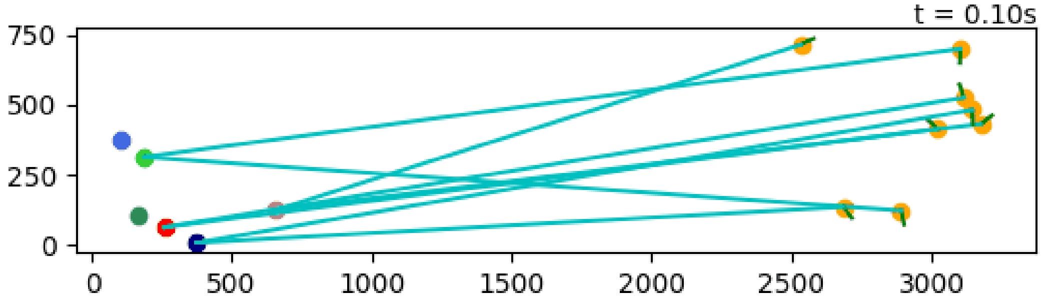


Results – Completion CF



Time (s)	Event
7.1	Agent 1 gets attrited
17.0	Agent 3 gets attrited
47.4	Agent 7 collides with Target 0; Target 0 survives
47.8	Agent 0 collides with Target 0; Target 0 is destroyed
56.9	Agent 5 collides with Target 1; Target 1 is destroyed
60.6	Agent 4 collides with Target 2; Target 2 is destroyed
63.9	Agent 6 collides with Target 3; Target 3 is destroyed
69.0	Agent 2 collides with Target 4; Target 4 survives





Conclusion/Future Work

- These three novel cost functions induce different behaviors in target assignment as expected and work reliably within the decentralized framework of the problem

Future Work

- Implement in 3D simulation environment
  - 3DOF and 6DOF dynamics models
  - Midcourse/terminal phase guidance laws
  - Defending agents replace random attrition
- Methods for estimating attrition probability and weapons effectiveness

Questions?