Adaptive Team Formation for Collaborative Perimeter Defense

Cleiton Neves Santos[1] **and Douglas G. Macharet**[1]

[1]Laboratório de Visão Computacional e Robótica (VeRLab)
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG – Brazil

{cleiton.ns,doug}@dcc.ufmg.br

Abstract. *In the Perimeter Defense Problem, teams of mobile robots must intercept intruders before they breach a protected region. This work introduces adaptive team formation and collaborative interception methods for both homogeneous and heterogeneous defender scenarios. Homogeneous teams use a Weapon-Target Assignment inspired network-flow formulation to optimize task assignment and routing, while heterogeneous defenders teams—differing in speed—are modeled as an Unsplittable Flow Problem and solved via a successive shortest-path heuristic. Experiments show that collaborative capture and strategic team composition significantly boost performance, and that the heuristic balances solution quality and speed, making it suitable for real-time security applications. Illustrative video: https://youtu.be/eB6NXVCknxk.*

Keywords: *Perimeter Defense, Multi-robot Systems, Flow Networks.*

MSc. thesis defended in 09/06/2025 (PPGCC/UFMG). Committee: Prof. Douglas G. Macharet (DCC/UFMG), Prof. Luiz Chaimowicz (DCC/UFMG) and Prof. Thiago F. Noronha (DCC/UFMG).

1. Introduction

With advancements in computational power and increasingly sophisticated algorithms, the feasibility of deploying robotic systems across diverse real-world applications has expanded substantially. Multi-Robot Systems (MRS) have emerged as a promising approach to address various challenges in domains that require distributed or autonomous solutions, like surveillance, environmental monitoring and area coverage. In this context, the need for adaptive and collaborative strategies within MRS becomes particularly evident in domains like Perimeter Defense [Shishika and Kumar 2020, Smith et al. 2009, Bopardikar et al. 2010, Bajaj and Bopardikar 2019, Macharet et al. 2020, Chen et al. 2021] (Figure 1), where dynamic environments require real-time decision-making and coordination. By enhancing collaboration and efficiency among robots in dynamic, time-sensitive environments, this work aims to optimize the coordination and adaptability of robotic defenders, enabling them to collaborate effectively, form teams dynamically, and adapt their strategies in response to intruders in a Perimeter Defense Problem (PDP) scenario.

We propose two collaborative strategies for the PDP. For homogeneous defenders, a flow network inspired by the Weapon-Target Assignment Problem

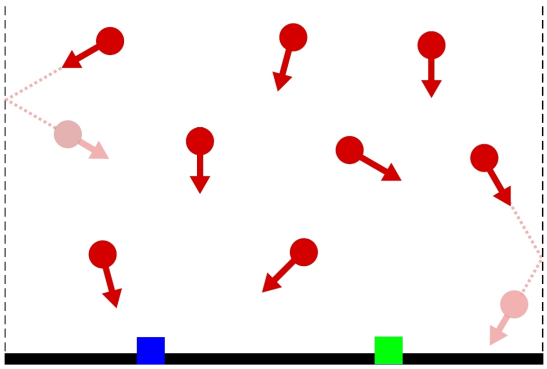


Figure 1. Defenders (squares) must intercept intruders (red circles) before they reach the perimeter (bottom black line). The lateral dashed lines indicate the boundaries of the environment, where intruders rebound upon contact.

(WTA) [Ahuja et al. 2007] model solves the Minimum-Cost Maximum-Flow (MCMF) problem, determining assignments and routes. For heterogeneous defenders, the problem is modeled as an Unsplittable Flow Problem (UFP) [Kleinberg 1996], addressing varying velocities. Both approaches enable adaptive team formations, reducing the likelihood of high-priority intruders escaping. Through experimentation and analysis, we demonstrate the effectiveness of our proposed approaches in enhancing defense performance in both homogeneous and heterogeneous settings. The insights gained from this research can inform the design and deployment of future autonomous defense systems, paving the way for more resilient and intelligent security solutions.

1.1. Contributions

This work makes the following key contributions:

- A novel approach to the Perimeter Defense problem that explicitly accounts for uncertainty in intruder capture and enables collaborative interceptions;
- A novel methodology for adaptive team formation and collaborative intruder interception in the Perimeter Defense Problem (PDP) within Multi-Robot Systems (MRS) based on network flow theory and the Weapon-Target Assignment Problem (WTA) [Ahuja et al. 2007] for efficient task allocation, routing, and coordination in homogeneous defender configurations.
- A model for heterogeneous defender configurations, where defenders differ in maximum velocity, formulated as an Unsplittable Flow Problem (UFP) and a successive shortest path heuristic to address the computational complexity of UFP, ensuring real-time applicability in heterogeneous scenarios.

2. Problem Formulation

Given an obstacle-free rectangular environment $\mathcal{E} \subset \mathbb{R}^2$, consider a line $\mathcal{P}$ on the bottom of $\mathcal{E}$, which is the perimeter. Intruders consistently enter the environment from the top edge, advancing linearly toward the bottom edge at a constant speed. A team of defenders is tasked with preemptively intercepting the intruders, ensuring they do not breach $\mathcal{P}$. We also consider that collisions among defenders can be ignored and depict defender collaboration as the convergence of defenders in a single point for an interception.

Definition 1 (Perimeter)**.** *Without loss of generality, we define a linear perimeter $\mathcal{P}$ to be located in the x-axis with length $w \in \mathbb{R}^+$. Using Cartesian coordinates, the perimeter is a line segment between the points (0, 0) and (w, 0).*

Definition 2 (Defenders)**.** *We assume a team of defenders $\mathcal{D} := \{\mathcal{D}_i \ \forall i \in \mathbb{N}^+ \mid i \leq N_d\}$ spatially distributed along $\mathcal{P}$. We represent each defender's position, in the Cartesian coordinates, by $\mathbf{p}_i^d = [x_i^d, y_i^d]^T$. Given the constraint that all defenders move only along $\mathcal{P}$, it is always the case that $y_i^d = 0$. Additionally, there is a speed limit $v_i^d \leq |\dot{x}_i^d|$, where the defenders' dynamics adhere to a first-order model, enabling direct control over its velocity. In this work, we disregard the collision among defenders.*

Definition 3 (Defenders' Collaboration)**.** *In this work, collaboration is the ability of multiple defenders to jointly attempt a capture, thereby increasing the overall probability of success in a probabilistic setting. For example, consider an intruder $\mathcal{A}_j$ with a capture evasion probability of q_j^a. If one defender is assigned, the probability of success is $1 - q_j^a$. With two defenders collaborating, the success probability becomes $1 - q_j^a + q_j^a(1 - q_j^a)$, which is greater than $1 - q_j^a$, demonstrating the benefit of collaboration. We also impose a constraint on the maximum number of defenders allowed to collaborate, denoted by the parameter $M \in \mathbb{N}^+$, to control and assess the impact of collaboration explicitly.*

Definition 4 (Intruders)**.** *An intruder $\mathcal{A}_j$ is an agent advancing towards $\mathcal{P}$. Its spatial coordinates are defined by $\mathbf{p}_j^a = [x_j^a, y_j^a]^T$ in Cartesian space. Each intruder moves with a velocity v_j^a and a trajectory angulation $\theta_j^a \in [\frac{5\pi}{4}, \frac{7\pi}{4}]$, such that its velocity components are given by: $\dot{x}_j^a = v_j^a \cos\theta_j^a, \quad \dot{y}_j^a = v_j^a \sin\theta_j^a$. Upon reaching the side boundaries of the environment, an intruder undergoes a horizontal rebound, preserving its speed but reversing the sign of its horizontal velocity component, i.e., $\dot{x}_j^a \leftarrow -\dot{x}_j^a$, while maintaining $\dot{y}_j^a$ unchanged, as illustrated in Figure 1. Each intruder is associated with a reward upon successful capture, represented by $r_j^a \in \mathbb{R}^+$, and possesses a probability of evading capture, denoted as q_j^a. Finally, we assume the number of current intruders in the environment at a given time is always known and denoted by N_a.*

Intruders are inserted in the environment sequentially over time, following a Poisson process with a rate denoted as λ. The horizontal displacement x_j^a of a new intruder $\mathcal{A}_j$ is defined by a uniform distribution, with a fixed initial vertical distance, $y_j^a = \eta$, from the perimeter $\mathcal{P}$. Upon arrival, each new intruder $\mathcal{A}_j$ initiates movement towards $\mathcal{P}$ with a constant velocity denoted as v_j^a.

A defender, $\mathcal{D}_i$, performs a capture attempt on an intruder, $\mathcal{A}_j$, when $||\mathbf{p}_i^d - \mathbf{p}_j^a|| = 0$. Notably, the capture attempts made by defenders are independent events. Consequently, $(q_j^a)^{n_j^a}$ denotes the probability of escape of an intruder $\mathcal{A}_j$, where n_j^a is the number of defenders assigned to perform a capture attempt on the specified intruder.

In summary, our problem can be defined as:

Problem 1 (Adaptive Team Formation for Perimeter Defense with Probabilistic Captures)**.** *Given an environment $\mathcal{E}$ with a well-defined perimeter $\mathcal{P}$ and a team consisting of N_d defenders freely to move within this perimeter. The objective is to formulate a collaborative defense strategy that maximizes expected intruder capture, considering intruders' priority.*

The methodology's performance is evaluated through the ratio between the expected capture of intruder rewards and the total expected reward of intruders, expressed as follows:

$$\mathcal{C} := \lim_{t \to +\infty} \mathbb{E}\left[\frac{r_{capt}(t)}{r_{total}(t)}\right] , \tag{1}$$

where $r_{\text{capt}}(t)$ is the intruder reward captured, and $r_{\text{total}}(t)$ is the total intruder reward, both at time t.

3. Perimeter Defense with Homogeneous Defenders

This section tackles the PDP using homogeneous defenders. All the defenders present identical characteristics, presenting the same maximum velocity. Intruders, on the other hand, are heterogeneous, each presenting a different reward and capture evasion probability; however, intruders are homogeneous in terms of speed. Intruders move radially toward the perimeter in a straight line, and such behavior is consistent with observed patterns in Critical Infrastructure Protection (CIP) contexts as shown in [Schneider et al. 2021] where a straight-line approach pattern is present among threats posed by intruders.

3.1. Graph Construction

The flow network formulation uses a directed graph $G = (V, E)$ with capacities $c(u, v)$ and costs $a(u, v)$ for each edge $e \in E$, and two distinct vertices $\mathbf{S}$ (source) and $\mathbf{T}$ (sink). The Minimum Cost Maximum Flow problem aims to find the maximum flow f from $\mathbf{S}$ to $\mathbf{T}$ that minimizes the total cost while satisfying the capacity constraints.

First, we describe how we created the initial graph (before the arrival of any intruders). We create nodes $\mathbf{S}$ and $\mathbf{T}$, respectively source and sink. We also insert distinct nodes, denoted $\mathbf{n}_i^d$, to represent each defender $\mathcal{D}_i$. Next, we add edges from $\mathbf{S}$ to all defender nodes $\mathbf{n}_i^d$ and, in this initial graph, from all defender nodes to $\mathbf{T}$. Edges have one unit of capacity and zero cost. Upon the arrival of intruders, it is necessary to update the initial graph by creating connections (edges) between defenders and intruders. These connections represent the potential for a defender to capture an intruder, and their establishment is contingent upon the concept of *reachability*.

Definition 5 (Reachability)**.** *An intruder $\mathcal{A}_j$ at position $\mathbf{p}_j^a = [x_j^a, y_j^a]^T$ is considered reachable by defender $\mathcal{D}_i$ at position $\mathbf{p}_i = [x_i^d, y_i^d]^T$ and velocity v^d, if the following is satisfied:*

$$\frac{|x_j^a(t) - x_i^d(t)|}{v^d} \leq \frac{y_j^a(t) - y_i^d(t)}{v_j^a}\,. \tag{2}$$

Figure 2 shows how the concept is applied to a simple scenario. In this context, an edge is created to represent an intruder is reachable, i.e., the defender can intercept it when it reaches the perimeter.

We update the initial graph when an intruder arrives or leaves as follows. For each new intruder, we create at most $M + 1$ nodes: one node $\mathbf{n}_j^{a_{in}}$ for the intruder and M nodes $\mathbf{n}_j^{a_{out\_i}}$ representing its potential assignments. Unitary capacity, zero-cost edges are added from all defender nodes $\mathbf{n}_i^d$ to $\mathbf{n}_j^{a_{in}}$, and from each $\mathbf{n}_j^{a_{out\_i}}$ to the sink ($\mathbf{T}$). If an intruder $\mathcal{A}_k$ is reachable from $\mathcal{A}_j$, edges are similarly created from each node $\mathbf{n}_k^{a_{out\_i}}$ to $\mathbf{n}_j^{a_{in}}$. Additional unit capacity edges are added from $\mathbf{n}_j^{a_{in}}$ to each corresponding $\mathbf{n}_j^{a_{out\_i}}$ with costs defined as

$$a(i, j) = -r_j^a\,(q_j^a)^{(i-1)}(1 - q_j^a).$$

For instance, if intruder $\mathcal{A}_v$ is reachable by any two defenders from the total N_d, the first edge cost is $a(1, v) = -r_v^a(1 - q_v^a)$ and the second is $a(2, v) = -r_v^a q_v^a(1 - q_v^a)$, ensuring

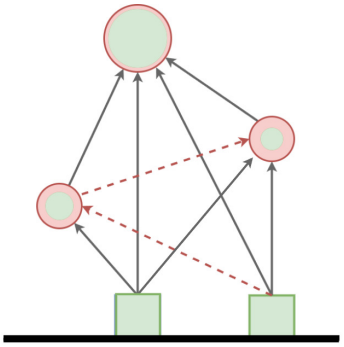


Figure 2. An arrow extending from one agent and pointing to another signifies that the second agent is reachable from the position of the first. Arrows with red dashed lines serve as examples where agents are unreachable.

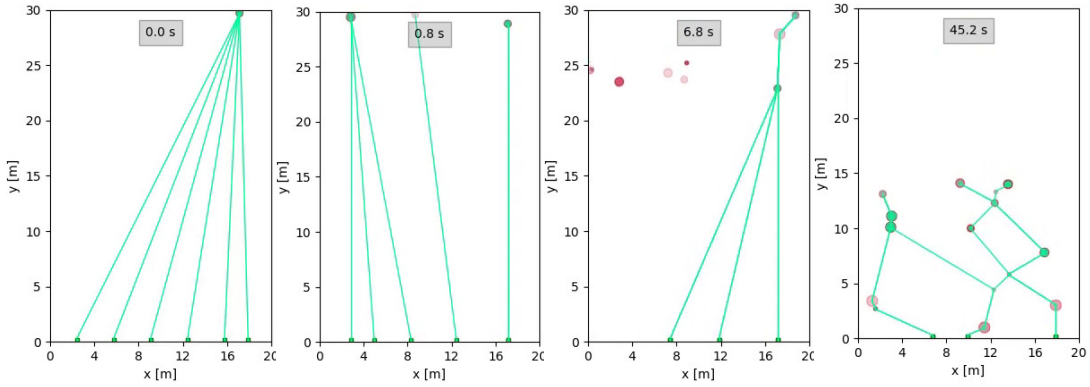


Figure 3. (a) First intruder within reach of all defenders; (b) assignment to distinct intruders; (c) reassignment and planning for subsequent captures; (d) adaptive evolution of assignments and routes.
Video: https://youtu.be/0ZjmfG3lIO8.

that sequential assignments yield lower total cost. This behavior arises from two factors: (1) the problem's cost-minimizing nature and (2) unit edge capacities enforcing distinct selections for defenders. Finally, when an intruder reaches the perimeter or is captured, its corresponding nodes and edges are removed.

3.2. Determining Assignments and Routing

A valid min-cost max-flow in the network described in Section 5 assigns each intruder $\mathcal{A}_j$ (via node $\mathbf{n}_j^{a_{in}}$) to a defender $\mathcal{D}_i$ by routing flow from the source through defender nodes to the sink. The route taken by a defender is determined by the order in which its corresponding intruder nodes are encountered in the flow, ensuring that the defender sequentially intercepts the intruders in accordance with the defined reachability criteria outlined in Definition 5.

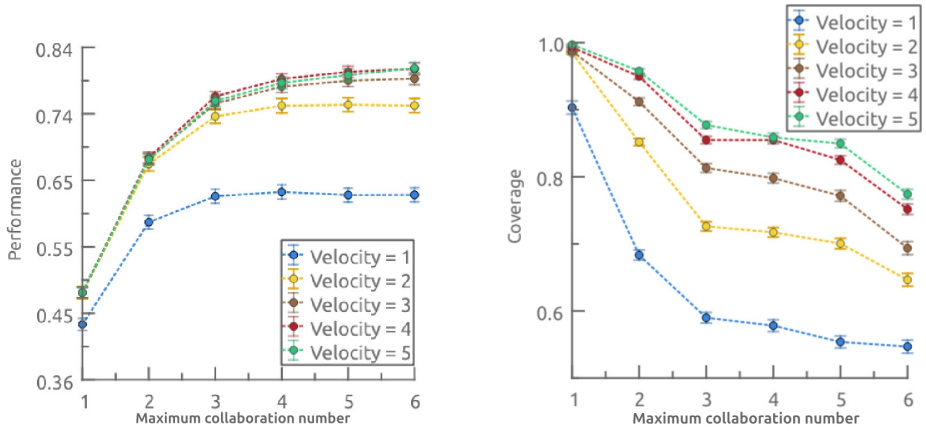
3.3. Experiments

We evaluated our approach in a simulated perimeter-defense scenario implemented using `NetworkX` and `NumPy` on Ubuntu22.04 (16GBRAM, AMD Ryzen 7 3700U). Each trial

deployed $N_d = 6$ defenders uniformly along a 20m perimeter (the x-axis of a 20m×10m arena). Intruder arrivals followed a Poisson process ($\lambda = 1$ over 30s), commencing 10m from the perimeter at 1m/s. Each intruder's evasion probability was drawn from $U(0.1, 0.9)$, with reward values in $\{1, 10, 100, 1000, 10000\}$ and uniform arrival positions along the y-axis. To clarify our method, we present a representative run in a 20m×30m arena. Defenders (green squares) patrol at $v^d = 1$m/s and may form teams of up to $M = 6$. Intruders appear as red circles whose size encodes capture reward; inner green circles indicate instantaneous capture probability. As defenders reassign and reroute, cooperative teams dynamically form to maximize success.

3.3.1. Capture Performance and Coverage

We analyzed how the maximum cooperation size M (from 1 to 6) and defender speed $v^d \in [1, 5]$m/s affect both system performance (as defined in Equation 1) and the coverage ratio (fraction of intruders targeted). Over 30 runs per M, higher M consistently boosts performance but reduces coverage, with diminishing returns beyond $M = 3$.



(a) (b)

Figure 4. Average performance (a) and coverage (b) vs. cooperation level M and defender velocity v^d, including standard error bars.

3.3.2. Velocity vs. Cooperation

A 2^2 factorial design (factors: defender speed $v^d \in \{1, 5\}$m/s and cooperation $M \in \{1, 5\}$), with 30 replicates per condition (120 runs total), revealed that cooperation (M) explains 54.18% of performance variance, velocity 10.42%, their interaction 3.49%, and residual error 31.91%. An F-test confirms these factors are significant at the 99% level, indicating that enabling defender collaboration has a substantially greater impact on overall performance than increasing speed.

4. Perimeter Defense with Heterogeneous Defenders

In this chapter, we tackle the PDP using heterogeneous defenders regarding maximum velocity. Intruders are heterogeneous in terms of reward, probability of escape, and

Table 1. Results of the 2^2 factorial experiment: v^d (m/s), cooperation M, minimum $\mathcal{C}_{min}$, maximum $\mathcal{C}_{max}$, mean $\overline{\mathcal{C}}$, and coefficient of variation (C.V.).

$\boldsymbol{v^d}$ (m/s)	$\boldsymbol{M}$	$\mathcal{C}_{min}$	$\mathcal{C}_{max}$	$\overline{\mathcal{C}}$	C.V.
1	1	0.1914	0.6922	0.4377	0.2196
5	1	0.3196	0.6924	0.4839	0.1703
1	5	0.3202	0.7944	0.6255	0.1750
5	5	0.5657	0.9577	0.8000	0.1211

velocity. They also exhibit different angulation upon arriving in the environment following a uniform distribution $\theta_j^a \sim U(\frac{5\pi}{4}, \frac{7\pi}{4})$ and rebound upon collision on the environment boundaries as illustrated in Figure 1. Such behavior is consistent with observed patterns in CIP contexts as demonstrated in [Schneider et al. 2021] where straight-line and zig-zag approach patterns are present among threats posed by intruders.

5. Graph Construction

The flow network formulation uses a directed acyclic graph $G = (V, E)$ with capacities $c(u, v)$ and costs $a(u, v)$ for each edge $e \in E$, and two vertices $\mathbf{S}$ (source) and $\mathbf{T}$ (sink).

First, we describe how we created the initial graph (before the arrival of any intruders). We create nodes $\mathbf{S}$ and $\mathbf{T}$, respectively source and sink. We also insert distinct nodes, denoted $\mathbf{n}_i^d$, to represent each defender $\mathcal{D}_i$. Next, we add edges from $\mathbf{S}$ to all defender nodes $\mathbf{n}_i^d$ and, in this initial graph, from all defender nodes to $\mathbf{T}$. Those edges have zero cost. Upon the arrival of intruders, it is necessary to update the initial graph by creating connections (edges) between defenders and intruders. These connections represent the potential for a defender to capture an intruder, and their creation are conditioned upon the updated concept of *reachability*.

Definition 6 (Reachability: Heterogeneous Velocity and Angulation)**.** *An intruder $\mathcal{A}_j$ at position $\mathbf{p}_j^a = [x_j^a, y_j^a]^T$ is considered reachable by defender $\mathcal{D}_i$ at position $\mathbf{p}_i = [x_i^d, y_i^d]^T$ and velocity v^d, if the following is satisfied:*

$$\frac{|x_{intercept}^{aj} - x_i^d(t)|}{v_i^d} \leq \frac{y_j^a(t) - y_i^d(t)}{|\dot{y}_j^a|} \,, \tag{3}$$

where $x_{\text{intercept}}^{a_j}$ denote the x-coordinate of intruder $\mathcal{A}_j$ at the instant it intersects the perimeter. This coordinate is defined as:

$$x_{\text{intercept}}^{a_j} = (-1)^{(n_{\text{reflect}} \bmod 2)} \cdot x_{\text{displac}} \mod |\mathcal{P}| \,, \tag{4}$$

with $t_{\text{displac}} = \frac{y_j^a(t)}{|\dot{y}_j^a|}$, representing the time required for intruder to reach the perimeter,
$x_{\text{displac}} = \left| x_j^a + t_{\text{displac}} \cdot \dot{x}_j^a \right|$,corresponding to the x-axis position at interception,
$n_{\text{reflect}} = \left\lfloor \frac{x_{\text{displac}}}{|\mathcal{P}|} \right\rfloor$, the number of reflections at the boundaries prior to interception.

Figure 5 illustrates the updated concept applied to a simple heterogeneous scenario. Once again, an edge is created to represent an intruder is reachable, *i.e.*, the defender can intercept it when it reaches the perimeter.

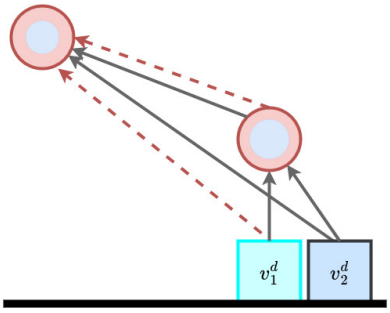


Figure 5. Example of reachability in a heterogeneous scenario. In the scenario illustrated above $\mathbf{v_1^d} < \mathbf{v_2^d}$. An arrow extending from one agent and pointing to another represents that the second agent is reachable from the first considering angulation, position and velocity of both. Arrows with red dashed lines serve as examples where agents are unreachable.

Subsequently, we describe the procedure for updating the initial graph given an intruder's arrival or leave.

We create at most $M + 1$ nodes to represent the new intruder, $\mathbf{n}_j^{a_{in}}$, and their M potential assignments, $\mathbf{n}_j^{a_{out\_i}}$. Edges are established with zero cost and capacity $c = 1/v_i^d$, extending from all defender nodes $\mathbf{n}_i^d$ to a reachable intruder node $\mathbf{n}_j^{a_{in}}$. Similarly, edges originate from all $\mathbf{n}_j^{a_{out_i}}$ nodes and lead towards the designated sink node ($\mathbf{T}$). Additionally, if an intruder $\mathcal{A}_k$ is deemed reachable from $\mathcal{A}_j$ for an defender $\mathcal{D}_i$, a connection is established from the $\mathbf{n}_k^{a_{out\_i}}$ node to $\mathbf{n}_j^{a_{in}}$ with capacity $c = 1/v_i^d$ and zero cost. We assess reachability among intruders to achieve the sequential capture behavior, as a defender (or multiple ones) assumes the same position as the intruder at the moment of a capture attempt. Next, edges with capacity $c = 1/v_i^d$ are generated from each node $\mathbf{n}_j^{a_{in}}$ to its corresponding node $\mathbf{n}_j^{a_{out\_i}}$.

We chose the capacities for the edges so that the defenders' velocity is considered, reflecting their capacity to reach specific intruders. Hence, intruder nodes will have connections only to defender nodes corresponding to defenders whose velocity can reach the intruder associated with the intruder node. The costs of these edges are determined by:

$$a(i, j) = -r_j(q_j)^{(i-1)}(1 - q_j) \, . \tag{5}$$

This calculation involves multiplying the probability of the i^{th} assigned defender effectively capturing intruder $\mathcal{A}_j$ by the associated reward r_j of the intruder and then by -1. Adjusting these edges' weights allows for adaptation of the formulation to various probability scenarios. Finally, when an intruder $\mathcal{A}_j$ either reaches the perimeter or is captured by a defender, its corresponding nodes $\mathbf{n}_j^{a_{in}}$ and $\mathbf{n}_j^{a_{out}}$, and respective edges, are removed from the graph.

6. Determining Assignments and Routing

As can be observed in the concept of reachability, the assignable intruders depend on the defender's speed. The reachability graph reflects this characteristic by setting different demands for each defender node and edge capacities, accounting for the defender's ability to reach the intruder. Our objective is that each demand depicts the path a defender must follow through the intruders. Hence, it should not split into many paths toward the sink.

Furthermore, two demands can not share an edge, as they must be used only once to account correctly for the costs associated with the result. We model the problem as an Edge-Disjoint UFP [Kleinberg 1996] to account for this new configuration. This section formulates the assignment and routing problem as an Edge-Disjoint Unsplittable Flow Problem. We then introduce a polynomial-time heuristic based on the successive shortest-path approach for obtaining solutions.

6.1. Edge-Disjoint Unsplittable Flow Formulation

The Unsplittable Flow Problem (UFP) [Kleinberg 1996] is a network optimization challenge in which each demand or commodity—defined by a source, a destination, and a demand size—must be routed entirely along one single path rather than being split over several paths. In contrast to splittable flow problems, where flows can split among multiple routes to best use the network's capacity, the UFP requires that each commodity's flow be unsplittable. The specific problem at hand also evokes the necessity of an edge-disjoint solution. Here, we model the problem as an Integer Linear Program (ILP), where each commodity's flow must route along a unique path without splitting. The following variables and parameters are defined:

- $\boldsymbol{x_{uv}^i} \in \{0, 1\}$: A binary decision variable indicating whether commodity i utilizes edge (u, v).
- $\boldsymbol{d_i}$: The flow demand size for commodity i, which is to be routed from source s_i to destination t_i.
- $\boldsymbol{a_{uv}}$: The capacity of edge (u, v).
- $\boldsymbol{c_{uv}}$: The cost associated with traversing edge (u, v).

Objective Function: Minimize the total routing cost:

$$\min \sum_{(u,v)\in E} \sum_{i=1}^{k} c_{uv} \cdot x_{uv}^i. \tag{6}$$

Capacity Constraints: The cumulative flow on each edge must not exceed its capacity:

$$\sum_{i=1}^{k} d_i \cdot x_{uv}^i \leq a_{uv}, \quad \forall (u, v) \in E. \tag{7}$$

Flow Conservation Constraints: For each commodity i and node u, the net flow is defined as:

$$\sum_{(u,v)\in E} x_{uv}^i - \sum_{(v,u)\in E} x_{vu}^i = \begin{cases} 1, & \text{if } u = s_i, \\ -1, & \text{if } u = t_i, \\ 0, & \text{otherwise,} \end{cases} \quad \forall u \in V, \quad \forall i \in \{1, \ldots, k\}. \tag{8}$$

Edge-Disjointness Constraint: To ensure that no two commodities share the same edge:

$$\sum_{i=1}^{k} x_{uv}^i \leq 1, \quad \forall (u, v) \in E. \tag{9}$$

Complexity Class: Although the UFP instance presented in this work is constrained to a directed acyclic graph structure, it still retains the intrinsic NP-hardness characteristic of the general UFP. Given a solution to the problem, the path for each demand to the sink defines the allocations through the correspondence between edges and intruders and the path that the corresponding defender must take throughout the intruders.

6.2. Successive Shortest Path Heuristic

We formulate a successive shortest path heuristic to address the NP-hardness of the problem. The method iteratively routes each demand by computing the shortest path—based on the defined cost metric—while updating the residual capacities of the network, offering a practical approach for large-scale instances.

Algorithm Steps

1. **Sort Demands by Size:** Reorder demands such that $d_1 \geq d_2 \geq \cdots \geq d_{N_d}$. *Intuition*: Demands require edges with $a(e) \geq d_i$, which get scarce as demands get larger. Routing them first reserves these critical edges, preventing smaller demands from occupying partial capacity and blocking paths for larger demands.
2. **Iterative Path Routing:** For each demand (s_i, t_i, d_i):
   (a) Construct a residual graph $\mathcal{G}_{\text{res}}$ by removing all edges e where $a(e) < d_i$.
   (b) Find a shortest path P_i in $\mathcal{G}_{\text{res}}$ using Bellman-Ford.
   (c) If P_i exists, set $a(e) \leftarrow 0$ for all $e \in P_i$, ensuring edge-disjointness.

7. Experiments

We conducted all simulations in Python 3 using `NetworkX` and `NumPy` on Ubuntu 22.04 (16GB RAM, AMD Ryzen 7 3700U). The domain is a 20m×30m rectangle with a 20m perimeter ($w = 20$) along the x-axis. Defenders begin equidistantly spaced on the perimeter. Intruder arrivals follow a Poisson process ($\lambda = 2$, 30 s window), each starting 30 m from the perimeter with speed $v^a \in \{1, 3, 5\}$ m/s. Capture rewards are drawn from $\{1, 10, 100, 1000, 10000\}$; escape probabilities follow $U(0.1, 0.9)$. Trajectories are angled $\theta_j^a \sim U(5\pi/4, 7\pi/4)$, with uniform y_j^a.

Defender speeds are heterogeneous: $\{2, 3, 3, 6, 6, 7\}$m/s, $M = 6$. Figure 6 shows defenders (green squares) forming dynamic teams to intercept intruders (red circles). Inner blue circles indicate instantaneous capture probability, and intruder size reflects capture reward. Faster defenders both extend coverage and enhance capture performance by joining more teams (see Section 7.1).

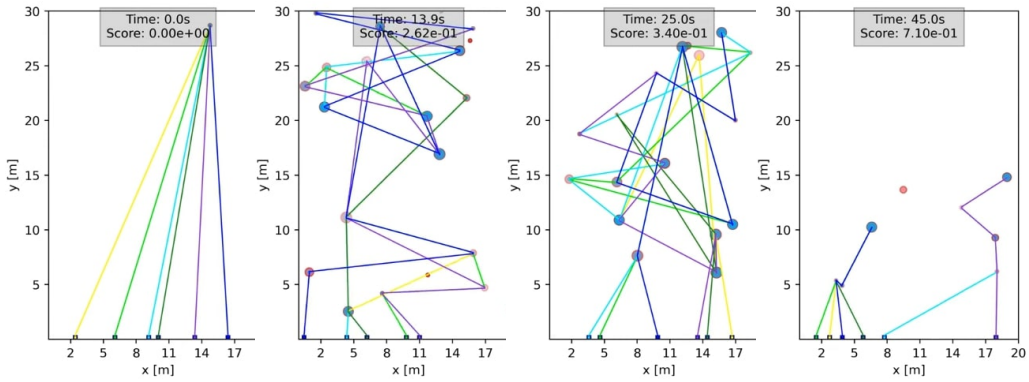
7.1. Impact of Defender Heterogeneity

We compare two configurations over 100 paired trials:

- **Heterogeneous:** speeds $\{2, 3, 3, 6, 6\}$m/s
- **Homogeneous:** all at $v^d = 4$m/s

All other parameters as in Section 7.

A paired t-test (Kolmogorov–Smirnov normality passed) shows heterogeneous defenders outperform homogeneous in coverage (99% confidence) and marginally in capture (77% confidence), demonstrating benefits of velocity diversity.



(a) (b) (c) (d)

Figure 6. (a) Initial coordinated response; (b) dynamic reassignment; (c)–(d) team formation and continuous routing adaptation.
Video: https://youtu.be/eB6NXVCknxk.

Table 2. Mean capture performance $\overline{\mathcal{C}}$ and coverage $\overline{\text{Coverage}}$ with C.V. over 100 paired runs.

Configuration	$\overline{\mathcal{C}}$	C.V.	$\overline{\text{Coverage}}$	C.V.
Homogeneous	0.7315	0.1121	0.6443	0.1032
Heterogeneous	0.7394	0.1096	0.7298	0.0843

7.2. Time-Efficiency and Capture Performance

We compare our heuristic to the LP-based optimal solution (PuLP + GLPK) over 100 paired runs, using UFP formulation (Section 6.1).

Solution Quality Wilcoxon signed-rank tests ($\alpha = 0.01$) reveal a mean performance gap of 0.0083 (99% CI [0.0016,0.015], $p < 0.01$) and coverage gap of 0.0052 (81% CI [0.0001,0.0102], $p = 0.19$).

Table 3. Optimal vs. heuristic: mean performance, coverage, and C.V. in 100 paired runs.

Method	$\overline{\mathcal{C}}$	C.V.	$\overline{\text{Coverage}}$	C.V.
Optimal	0.7397	0.1021	0.7340	0.0804
Heuristic	0.7394	0.1096	0.7298	0.0843

Scalability We tested on graphs of increasing size ($|V|, |E|$ from (36,106) up to (229,3382)), 100 runs each. The experiment results shown that the heuristic's markedly lower and more consistent runtimes compared to the LP. Overall, the heuristic offers acceptable sub-optimality in exchange for substantial computational gains.

8. Conclusions and Future Work

This thesis has presented a comprehensive approach to the Perimeter Defense Problem (PDP) within the context of Multi-Robot Systems (MRS), focusing on adaptive team formation and collaborative intruder interception. We have developed and validated methodologies for both homogeneous and heterogeneous defender configurations, demonstrating their effectiveness in enhancing defense performance in dynamic and unpredictable environments.

Several promising avenues for future research include developing adaptive collaboration policies that adjust to intruder density and defender availability, exploring broader forms of agent heterogeneity such as sensing and energy capacities, and investigating decentralized strategies under communication constraints to enhance the robustness and scalability of multi-agent defense systems.

References

Ahuja, R. K., Arvind Kumar, K. C. J., and Orlin, J. B. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55(6):1136–1146.

Bajaj, S. and Bopardikar, S. D. (2019). Dynamic boundary guarding against radially incoming targets. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4804–4809.

Bopardikar, S. D., Smith, S. L., Bullo, F., and Hespanha, J. P. (2010). Dynamic vehicle routing for translating demands: Stability analysis and receding-horizon policies. *IEEE Transactions on Automatic Control*, 55(11):2554–2569.

Chen, A. K., Macharet, D. G., Shishika, D., Pappas, G. J., and Kumar, V. (2021). Optimal Multi-robot Perimeter Defense Using Flow Networks. In *Proc. of 15th International Symposium Distributed Autonomous Robotic Systems (DARS)*, pages 282–293.

Kleinberg, J. M. (1996). Single-source unsplittable flow. *Proceedings of 37th Conference on Foundations of Computer Science*, pages 68–77.

Macharet, D. G., Chen, A. K., Shishika, D., Pappas, G. J., and Kumar, V. (2020). Adaptive Partitioning for Coordinated Multi-agent Perimeter Defense. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'20)*.

Schneider, M., Lichte, D., Witte, D., Gimbel, S., and Brucherseifer, E. (2021). Scenario analysis of threats posed to critical infrastructures by civilian drones. In *Proceedings of the 31st European Safety and Reliability Conference (ESREL 2021)*. Research Publishing Services.

Shishika, D. and Kumar, V. (2020). A review of multi agent perimeter defense games. In Zhu, Q., Baras, J. S., Poovendran, R., and Chen, J., editors, *Decision and Game Theory for Security*, pages 472–485, Cham. Springer International Publishing.

Smith, S. L., Bopardikar, S. D., and Bullo, F. (2009). A dynamic boundary guarding problem with translating targets. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8543–8548.