

Solving the Right Problem with Multi-Robot Formations

Chaz Cornwall

Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA

Jeremy P. Bos

Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA

Abstract

Formation control simplifies minimizing multi-robot cost functions by encoding a cost function as a shape the robots maintain. However, by reducing complex cost functions to formations, discrepancies arise between maintaining the shape and minimizing the original cost function. For example, a Diamond or Box formation shape is often used for protecting all members of the formation. When more information about the surrounding environment becomes available, a static shape often no longer minimizes the original protection cost. We propose a formation planner to reduce mismatch between a formation and the cost function while still leveraging efficient formation controllers. Our formation planner is a two-step optimization problem that identifies desired relative robot positions. We first solve an unconstrained problem to estimate non-linear and non-differentiable costs with a weighted sum of surrogate cost functions. We theoretically analyze this problem and identify situations where weights do not need to be updated. The weighted, surrogate cost function is then minimized using relative positions between robots. The desired relative positions are realized using a non-cooperative formation controller derived from Lyapunov's direct approach. We then demonstrate the efficacy of this approach for military-like costs such as protection and obstacle avoidance. In simulations, we show a formation planner can reduce a single cost by over 75%. When minimizing a variety of cost functions simultaneously, using a formation planner with adaptive weights can reduce the cost by 0-40%. Formation planning provides better performance by minimizing a surrogate cost function that closely approximates the original cost function instead of relying on a shape abstraction.

Introduction

Teams are often more effective than individuals when working towards complex objectives [1]. For mobile robot teams, many objectives can be reached by only using the relative, physical relationships between themselves and other entities in the environment. Examples of these objectives include entertainment [2], enclosing targets [3, 4], covering targets [5], social interaction [6], protecting payloads [7], surviving as a group [8], traffic control [9], deformable object transport [10], and others. The physical relationships that solve these types of problems are known as formations.

Mathematical tools for modeling formations first came from computer graphics [11] and biology [12]. One of the first applications of formations was aircraft and spacecraft maintaining a given shape for remote sensing applications [13, 14]. Formations in unpredictable and dynamic environments are now receiving more attention as robotic

technology matures [15, 16]. However, higher level goals, such as social acceptance or survival, often involve maintaining many physical relationships that cannot be described by a single shape. Formation planners attempt to optimize the relationship between physical position and higher level goals, but often rely on overly restrictive constraints [17, 18] or coarse discretizations [19, 20].

Our contributions are the following:

1. A formation planner that directly solves for a formation's shape using a cost function that closely aligns with the agents' objectives
2. A theorem and experimental evidence that describe the limitations of adaptive formation planning
3. A non-cooperative formation controller using Lyapunov's approach to implement any arbitrary formation
4. Multi-robot simulations showing the generality of our formation planner to different cost functions

Background

Let the pose of agents and other physical entities in the environment be known within an acceptable tolerance. A configuration is the spatial arrangement of the agents and physical entities in the environment, while a formation adds the additional constraint of having relationships between the agents and other physical entities. Figure 1 shows the distinction between a configuration and a formation.

We define an ideal formation as

$$X^*[t+1] = \arg \min_{X[t]} \sum_j r_j^*[t], \quad (1)$$

where $X[t]$ is the formation and $r_j^*[t]$ is the perfect quantification of the j th cost (i.e. property damage, safety violations, etc.). The cost function in (1) is the *true cost function*. Equation 1 usually cannot be implemented because a perfect representation of the cost does not exist. Practically, (1) is replaced by

$$\hat{X}^*[t+1] \approx \arg \min_{X[t]} \sum_j \hat{r}_j[t], \quad (2)$$

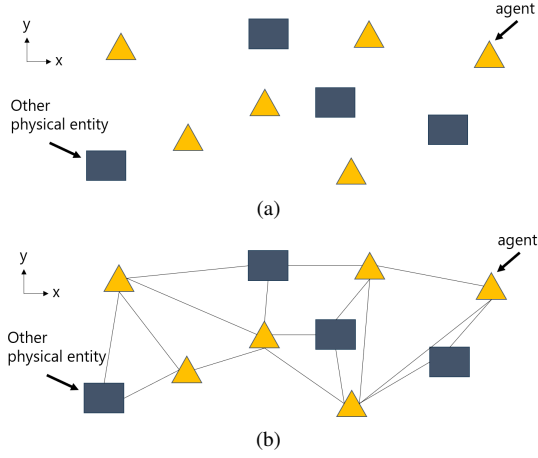


Figure 1: Difference between a configuration and formation. (a) Example configuration in $\mathbb{R}^2$. (b) Example formation in $\mathbb{R}^2$ with relative connections.

with the $\hat{\cdot}$ symbol denoting estimates. The cost function in (2) is the *surrogate cost function*.

Let $S[t]$ be non-controllable physical entities in the environment, $\mathbf{u}[t]$ the control signal, and $\mathbf{w}[t]$ the noise. Formation control calculates an input

$$\hat{\mathbf{u}}[t] = \arg \min_{\mathbf{u}[t]} d(X_d[t+1], g(X[t], S[t], \mathbf{u}[t], \mathbf{w}[t])) \quad (3)$$

to drive the current formation to a desired formation:

$$X_d[t+1] \approx g(X[t], S[t], \hat{\mathbf{u}}[t], \mathbf{w}[t]), \quad (4)$$

where $d(\cdot, \cdot)$ is a distance metric and the function $g(\cdot)$ models the dynamics of the agents.

The goal of a *formation system* is to have the controllable agents attain the ideal formation. This requires two steps: planning the ideal formation (2) and controlling the ideal formation (4). The main distinction between formation planners and formation controllers is the metric for finding the best formation. Formation planners use estimates of qualitative goals and formation controllers use distance from an ideal formation known *a priori*.

Current and Past Approaches

Formation Planning

Formation planning is conducted by humans or mathematical optimization. In most formation systems, humans select a formation *a priori* to give the formation controller. This type of formation planning assumes the formation shape given by the human is ideal by some metric. For soldiers and manned vehicles, the U.S. Army developed a formation planning system that considers the limitations and strengths of human soldiers and commanders [21]. The system outlines seven different formations that a commander can select: Column, Line, Wedge, Echelon, Vee, Diamond, or Box.

In affine formation control, new formations are derived from the locations of multiple leaders by identifying the affine transform that would be required to model the new positions of the leaders [17, 18]. Since the affine transformation is defined by leaders, who are not autonomously controlled, humans also determine the formation in this example. The underlying assumption in affine formation control is the ideal formation is an affine transformation of the original given shape, which is easily violated when the ideal formation is dependent on non-cooperative entities in the environment.

Optimizing for a formation using the agents' objectives is a new approach to formation planning. Instead of relying solely on human judgment, this type of formation planning determines the optimal formation shape implicitly defined in a cost function. ARL's TBAM-CRP has funded a few possibilities for military applications [22]. Oughourli, Limbu, and Ju implicitly represent formations in a Joint State Graph (JSG) where actions between robot locations are represented as edges and the cost of each edge factors in agent support [19]. Dimmig et al. formulate formations as mixed-integer linear programming (MILP) [20] to model a "bounding overwatch" behavior, which consists of robots watching over another group of robots. These approaches directly optimize for cost, but use graphs with discrete locations as the planning domain. Optimization over discrete states leaves the positions of the robots between discrete locations unanalyzed, jeopardizing mission success in dynamic environments.

Formation Control

The three paradigms of formation control are leader-follower, virtual structure, and behavior-based. Leader-follower methods cast formation control as a tracking problem: the follower just needs to maintain a formation representation with respect to the leader. Leader-follower approaches usually include one leader [23]. When more than one-leader is involved, agents are more cooperative and virtual structures become apparent [14, 18]. Instead of a leader deciding how the formation moves, a virtual structure or shape is created for the agents to cooperatively achieve. Common examples of virtual structure methods are multi-agent path planners that use the formation as a constraint [24, 10]. Behavior-based formation control uses a summation of behaviors to determine control signals for agents [25, 26, 27]. Behavior-based formation control is not currently a popular research area due to the difficulty in defining the behaviors and potential functions [28].

Given any of these paradigms, formations are implemented by using the desired shape as a constraint [29, 30] or supervisory signal for learning [31, 32, 5, 33]. However, as shown in Section 2, achieving the desired formation does not guarantee the minimum true cost.

Multi-Objective Optimization in Formations

Formation planning and control are both often represented as multi-objective optimization problems. The most used and intuitive method for multi-objective optimization is linear scalarization, which minimizes a weighted sum of the individual cost functions [34]. These weights control trade-offs and are selected according to human experience. For example, Lopez-Nicolas et al. suggest using α to weight one cost and $1 - \alpha$ to weight another cost, balancing between the preservation of scale or angles of the formation's shape [4]. Alonso-Mora et al. also use linear scalarization to calculate the optimal formation by weighting different shape features, such as distance to the goal and desired orientation [35]. However, by assuming the scalarization weights are constant and given as design parameters, these methods will have more difficulty adapting to changing objectives. We relax this assumption by approximating the true cost with a linear combination of relatively simple surrogate cost functions.

Methods

To ensure the formation system is solving the right problem, the formation planner optimizes a surrogate cost function that closely matches the true cost function. Changing the weights in the surrogate cost function through constrained least squares makes the match closer. Given the estimated ideal formation from the formation planner, a formation controller derived from Lyapunov's direct method then drives the agents to the formation shape. Figure 2 shows the interaction of the planner and controller in our formation system.

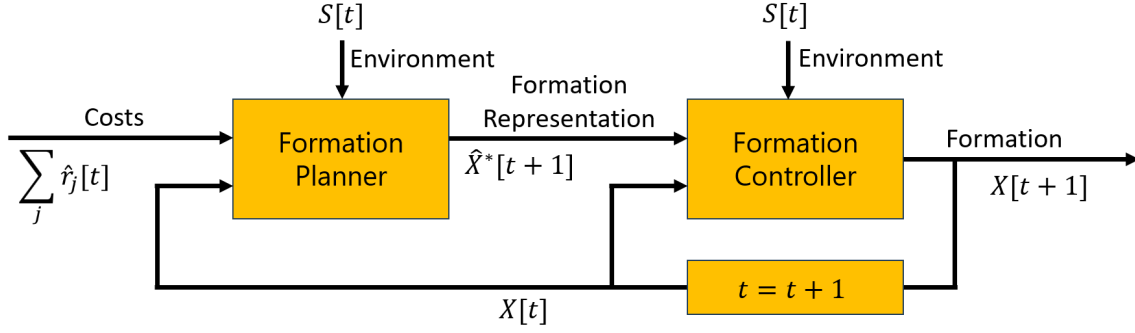


Figure 2: Diagram showing the elements of the formation system.

Formation Planner

The formation planner is a two stage process: fitting a surrogate cost function to the true cost function and optimizing for the estimated ideal formation.

Adaptive Weights for Fitting the Surrogate Cost

Let

$$c(\mathbf{x}) = \sum_{i=0}^{N-1} \alpha_i f_i(\mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{f}(\mathbf{x}) \quad (5)$$

be a scaled summation of functions where $c(\mathbf{x})$ is the surrogate cost function, and $\boldsymbol{\alpha}^T \mathbf{f}(\mathbf{x})$ is a linear combination of basis functions. Since (5) is a linear combination, the parameter N determines the dimension of the basis of c . When $N = 1$, $\boldsymbol{\alpha}$ has a unique solution. However, the weights $\boldsymbol{\alpha}$ are under determined whenever $N > 1$ because $\text{rank}(\mathbf{f}(\mathbf{x})) = 1$.

Knowing $\boldsymbol{\alpha}$ when $N = 1$ is not helpful because scaling a function by a positive value does not affect its maximum or minimum. However, for $N > 1$, c lies in the span of multiple basis functions $\{f_i\}$. Changing the weights will usually change local solutions of c , but there do exist conditions where this is not the case.

Theorem 1. Let $c_\alpha(\mathbf{x}) = \sum_{i=0}^{N-1} \alpha_i f_i(\mathbf{x})$ and $c_\beta(\mathbf{x}) = \sum_{i=0}^{N-1} \beta_i f_i(\mathbf{x})$ where $f_i(\mathbf{x}) \in C^1$. Let $A = [\nabla_x f_0(\mathbf{x}) \dots \nabla_x f_{N-1}(\mathbf{x})]$. Let $\hat{\mathbf{x}}_\alpha$ and $\hat{\mathbf{x}}_\beta$ be local solutions to $\min_{\mathbf{x} \in D_x} c_\alpha(\mathbf{x})$ and $\min_{\mathbf{x} \in D_x} c_\beta(\mathbf{x})$, respectively. Let γ be a scalar.

$$\exists(\boldsymbol{\alpha}, \boldsymbol{\beta}) : \hat{\mathbf{x}}_\alpha = \hat{\mathbf{x}}_\beta, \boldsymbol{\alpha} \neq \gamma \boldsymbol{\beta} \iff \det(A) = 0 \quad (6)$$

Proof. Assume $\hat{\mathbf{x}}_\alpha = \hat{\mathbf{x}}_\beta$. Using the first order necessary conditions for optimality [36],

$$\nabla_x c_\alpha(\mathbf{x}) = g(\mathbf{x}) \nabla_x c_\beta(\mathbf{x}) \text{ s.t. } g(\mathbf{x}) > 0, \mathbf{x} \in D_x. \quad (7)$$

Since $\nabla_x c_\alpha(\mathbf{x})$ and $\nabla_x c_\beta(\mathbf{x})$ are both elements in $\text{span}\{\nabla_x f_i(\mathbf{x})\}$, $g(\mathbf{x})$ must be a positive constant. Therefore, (7) simplifies to

$$\sum_i \nabla_x f_i(\mathbf{x}) [\alpha_i - g \beta_i] = 0 \implies \det(A) = 0, \quad (8)$$

which shows necessity. For sufficiency, assume $\det(A) = 0$. This can be expressed as

$$\sum_i \nabla_x f_i(\mathbf{x}) w_i = 0 \text{ s.t. } \mathbf{w} \neq \mathbf{0}. \quad (9)$$

If $w_i = [\alpha_i - g \beta_i]$, then (9) will satisfy (7) as long as g is selected to be positive. Since $\mathbf{w} \in \mathbb{R}^{N \times 1}$ and there would be $2N + 1$ unknowns in solving for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and g , the system is under-determined and there are infinitely many solutions. Therefore, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and g can be arbitrarily selected to satisfy $w_i = [\alpha_i - g \beta_i]$, resulting in (7). Since local solutions exist, we can imply from (7) that $\hat{\mathbf{x}}_\alpha = \hat{\mathbf{x}}_\beta$, showing sufficiency. $\square$

Theorem 1 shows changing weights will always change the minimum of the linear combination if and only if $\{\nabla_x f_i(\mathbf{x})\}$ is linearly independent. To ensure weight adaptation is effective, according to Theorem 1, we assume the gradients of the basis functions are linearly independent.

To model the true cost using the surrogate cost, we assume the true cost can be evaluated or observed independently from the basis functions $f_i(\mathbf{x})$, giving

$$c^*(\mathbf{x}) = c(\mathbf{x}) + e(\mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{f}(\mathbf{x}) + e(\mathbf{x}) \quad (10)$$

where $e(\mathbf{x})$ is the error between the observed true cost $c^*(\mathbf{x})$ and the surrogate cost $\boldsymbol{\alpha}^T \mathbf{f}(\mathbf{x})$. With additional data, the under-determined systems in (5) and (10) can be used to find the weights $\boldsymbol{\alpha}$ using constrained least squares. Let $\mathbf{c}_t(\mathbf{x}) \in \mathbb{R}^{M_t \times 1}$ and $F_t(\mathbf{x}) \in \mathbb{R}^{M_t \times N}$ where $M > N$ and

$$F_t(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}) \\ \vdots \\ \mathbf{f}_{M_t}(\mathbf{x}) \end{bmatrix}. \quad (11)$$

The optimization problem can be framed as

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_t &= \arg \min_{\boldsymbol{\alpha}} \|\mathbf{F}_t(\mathbf{x}) \boldsymbol{\alpha} - \mathbf{c}_t^*(\mathbf{x})\|_W^2 \\ \sum_{i=0}^{N-1} \alpha_i &= 1 \\ \boldsymbol{\alpha} &> \mathbf{0} \end{aligned} \quad (12)$$

with W as a diagonal weighting matrix. This is solved using the interior-point optimization library IPOPT [37]. The optimization problem in (12) is similar to the DONE algorithm [38] and dynamic movement primitives (DMP) [39], which both model more complex functions using weighted, linear combinations of basis functions. The approach here is unique due to the additional constraint of $\boldsymbol{\alpha} > \mathbf{0}$, which ensures the surrogate cost function does not become ill-formed for optimization.

Naively updating the cost function (12) will increase the size of $F_t(\mathbf{x})$ and $\mathbf{c}_t^*(\mathbf{x})$ at each iteration. Instead, the cost function is expanded and one-rank updates are used:

$$\alpha^T \underbrace{F_t^T W F_t}_{R_t} \alpha + 2\alpha^T \underbrace{F_t^T W \mathbf{c}_t^*}_{\mathbf{p}_t} + \underbrace{(\mathbf{c}_t^*)^T W (\mathbf{c}_t^*)}_{y_t} \quad (13)$$

where

$$R_t = \sum_{i=1}^{M_t} w_i \mathbf{f}_i \mathbf{f}_i^T = w_t (R_{t-1} + \mathbf{f}_{M_t} \mathbf{f}_{M_t}^T), \quad (14)$$

$$\mathbf{p}_t = \sum_{i=1}^{M_t} w_i \mathbf{f}_i \mathbf{c}_i^* = w_t (\mathbf{p}_{t-1} + \mathbf{f}_{M_t} \mathbf{c}_{M_t}^*) \quad (15)$$

and

$$y_t = \sum_{i=1}^{M_t} w_i \mathbf{c}_i^* \mathbf{c}_i^* = w_t (y_{t-1} + \mathbf{c}_{M_t}^* \mathbf{c}_{M_t}^*). \quad (16)$$

The weights w_i on the diagonal of W can be explicitly expressed as

$$w_i = \prod_{k=1}^{M_t} w_k \quad (17)$$

with $w_t \in [0, 1]$. This exponential weighting places higher importance on recent data than older data.

Optimizing for the Ideal Formation

After identifying the optimal weights, the formation planner uses the Adam optimizer [40] to find an estimate of the ideal formation. This optimizer is selected due to its prevalence in the deep learning community and its ability to minimize stochastic cost functions. Furthermore, since it does not rely on Hessians or estimates of Hessians, the optimizer performs well when the cost function is not strictly convex. From (2), the formation planner minimizes the surrogate cost:

$$\hat{\mathbf{x}}^* = \arg \min_{\mathbf{x}} \sum_{i=0}^{N-1} \hat{\alpha}_i f_i(\mathbf{x}) \quad (18)$$

where each function $f_i(\mathbf{x})$ is a cost the formation system may want to optimize and $\mathbf{x}$ represents the controllable entities (agents) in the environment.

Implementation

By assuming adaptive weights in the surrogate cost function, weights can change quickly or slowly depending on the speed and position of the agents with respect to the environment, constraining how much data we can use to estimate the true cost at a moment in time. Additionally, to obtain a good estimate of the weights, we need as much data as possible. Algorithm 1 manages this trade-off between too much and too little data to ensure (18) finds a near-ideal formation.

Algorithm 1 Formation Planning with Adaptive Weights

Input: $c_t^*, c_{t-1}^*, c_t, \mathbf{x}_t, \mathbf{x}_{t-1}, a_t, w_t, \tau_c, \tau_x, \tau_e$
1: **if** a_t and $d_c(c_t^*, c_{t-1}^*) > \tau_c$ and $d_x(\mathbf{x}_t, \mathbf{x}_{t-1}) > \tau_x$ **then**
2: $R_t, \mathbf{p}_t, y_t \leftarrow (14), (15), (16)$
3: $\hat{\alpha} \leftarrow (12)$
4: $i \leftarrow i + 1$
5: **if** $i > N$ and $\|F_t(x)\alpha - \mathbf{c}_t(x)\|_2^2 < \tau_e$ **then**
6: $\alpha \leftarrow \hat{\alpha}$
7: **end if**
8: **end if**
9: $\hat{\mathbf{x}}_{t+} \leftarrow (18)$
10: $c_{t+1} \leftarrow (5)$
11: Obtain c_{t+1}^*
12: **if** $c_{t+1}^* \geq c_t^*$ and $c_{t+1} \leq c_t$ **then**
13: $w_{t+1} \leftarrow U(l_w, u_w)$
14: $\tau_x \leftarrow U(l_{\tau_x}, u_{\tau_x})$
15: $a_{t+1} \leftarrow \text{True}$
16: **else**
17: $a_{t+1} \leftarrow \text{False}$
18: **end if**
19: **return** $a_{t+1}, w_{t+1}, \tau_x, \hat{\mathbf{x}}_{t+}$

Line 1 in Algorithm 1 reduces over-fitting by only using data $(c_t, \mathbf{x}_t)$ that is unique from the previous step. Line 5 tests the new weights to determine if they should be used. The condition $i > N$ ensures that at least enough data has been obtained so that the solution is not under-determined. The validation phase, occurring on line 12, checks whether the true cost decreases when optimizing over the surrogate cost. If the contrary is true, the data weighting parameter w_t (the diagonal elements of W) and the distance threshold τ_x are updated according to a random draw from a uniform distribution. The indicator a_t ensures new surrogate cost weights α are not re-estimated if the current estimates are effective.

Formation Control

Using Lyapunov's direct approach, a non-cooperative formation controller is developed that considers the velocities of the physical entities in the formation and the desired displacement between those entities. Let $\mathbf{x}_i$ be the i th controllable physical entity in the environment, and let $\mathbf{s}_j$ be the j th uncontrollable physical entity in the environment. Let the controllable physical entities have single-integrator dynamics: $\dot{\mathbf{x}}_i = \mathbf{u}_i$. Let

$$\begin{aligned} \mathbf{d}_{ij}^x &= \mathbf{x}_i - \mathbf{x}_j \\ \mathbf{d}_{ij}^s &= \mathbf{x}_i - \mathbf{s}_j \end{aligned} \quad (19)$$

be the desired displacements for neighboring entities. Let the errors be

$$\begin{aligned} \mathbf{e}_{ij}^x &= \mathbf{d}_{ij}^x - \hat{\mathbf{d}}_{ij}^x \\ \mathbf{e}_{ij}^s &= \mathbf{d}_{ij}^s - \hat{\mathbf{d}}_{ij}^s \end{aligned} \quad (20)$$

where $\hat{\mathbf{d}}_{ij}$ denotes a desired displacement between entities. The Lyapunov function can be expressed in terms of the errors as

$$V(X, S) = \frac{1}{2} \sum_i \left[\sum_{j \in \mathcal{N}_i^x} (\mathbf{e}_{ij}^x)^T \mathbf{e}_{ij}^x + \sum_{j \in \mathcal{N}_i^s} (\mathbf{e}_{ij}^s)^T \mathbf{e}_{ij}^s \right] \quad (21)$$

with $\mathcal{N}_i^x$ as the neighbors of i that are controllable and $\mathcal{N}_i^s$ as the neighbors of i that are not controllable. Since (21) is positive definite, the controller will be asymptotically stable if $\mathbf{u}_i$ is selected such that $\dot{V}(X, S) \leq 0$. Using chain rule,

$$\dot{V}(X, S) = \sum_i \left[\sum_{j \in \mathcal{N}_i^x} (\mathbf{e}_{ij}^x)^T (\mathbf{u}_i - \mathbf{u}_j) + \sum_{j \in \mathcal{N}_i^s} (\mathbf{e}_{ij}^s)^T (\mathbf{u}_i - \dot{\mathbf{s}}_j) \right]. \quad (22)$$

To minimize $\dot{V}(X, S) \leq 0$, we select the control signals such that

$$\begin{aligned} -\mathbf{e}_{ij}^x &= \mathbf{u}_i - \mathbf{u}_j \\ -\mathbf{e}_{ij}^s &= \mathbf{u}_i - \dot{\mathbf{s}}_j \end{aligned} \quad (23)$$

. The conditions in (23) can be shown for all the edges of the graph representing the formation using the incidence matrix B :

$$\begin{bmatrix} \mathbf{d}^x \\ \mathbf{d}^s \end{bmatrix} - B^T \begin{bmatrix} \mathbf{x} \\ \mathbf{s} \end{bmatrix} = B^T \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{s}} \end{bmatrix} \quad (24)$$

Letting $B = [B_1 \ B_2]$ and rearranging:

$$\begin{aligned} B_1^T (\mathbf{u} + \mathbf{x}) &= \mathbf{d} - B_2^T (\mathbf{s} + \dot{\mathbf{s}}) \\ B_1^T \mathbf{u} &= \mathbf{d} - \hat{\mathbf{d}} - B_2^T \dot{\mathbf{s}} \end{aligned} \quad (25)$$

Using (25), the control signal $\mathbf{u}$ can be calculated using any linear solver. If the linear system in (25) is over-determined, least-squares optimization should be used instead.

Experiment and Results

We evaluate our approach using simulations in completely known environments. Simulations using a protection scenario in several different environments show using our formation planner decreases a specific cost by 75% relative to formation systems without a formation planner. Experiments also show formation planning can decrease the cost over a set of different true cost functions by 20-40% relative to other formation systems.

Test Environment and Setup

All trials are executed within the Robotarium [41] simulator provided by Georgia Tech. There are four methods: Formation Planning with Adaptive Weights (FP-AW), Formation Planning with Static Weights (FP), Formation Shape (FS), and Affine Formation (AF). The FP approach selects a random set of weights and then uses those weights for the duration of each trial. This models the linear scalarization used for multi-objective optimization that is most common in the formation literature. The AF approach uses a popular technique called affine formation control [18]. In affine formation control, the formation shape is allowed to change as long as the current shape is an affine transformation of the desired shape. Table 1 shows the components in each formation system. The FS and AF methods also utilize barrier certificates (BC) to avoid entities in the environment [41]. Avoiding entities while using the FP-AW and FP methods is handled by the formation planner.

Table 1: Components of the Formation Systems Used in Experiments

Method	Formation Planner	Formation Controller
FP-AW	Numerical Optimization with Adaptive Weights	Section 4.2
FP	Numerical Optimization	Section 4.2
FS	Human	Section 4.2+BC
AF	Human	[18]+BC

There are four different types of physical entities: leader, agents, threats, and obstacles. The positions of all physical entities are

known. All obstacles and threats are static. Five different trials are conducted in each of the 4 environments using each technique (80 trials total). The FP method uses a different set of random weights for each trial. Figure 3 shows each environment’s initial conditions and the five waypoints the leader will follow. Each experiment is ran for 3000 steps or about 100 seconds. In situations where the leader reaches the last waypoint before the simulation has ended, the leader will return to the first waypoint and start again. The shape assigned to the FS and AF methods is a box, each agent at a corner and the leader in the middle.

True Costs

Three different types of true cost are used: protection cost, obstacle cost, and violation cost. The first two costs are continuous and twice-differentiable. The violation cost is not continuous. These costs represent the function $c^*(\mathbf{x})$ from (10) or quantities a user may care about that are more difficult to analyze than the basis functions in the surrogate cost. In a real-world application, the true cost only needs to be evaluated: first- and second-order information is assumed to be unavailable. Despite showing exact formulations of the true costs used in our experiments, the formulation planner only observes evaluations of these functions.

Some of the costs described in Section 5.2 and 5.3 are a function of the angle between two entities in the formation. Let the cosine of the angle between two vectors be represented as a *line-of-sight* (*los*) cost:

$$\text{los}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} = \cos \theta_{a,b}. \quad (26)$$

Equation 26 is also considered the cosine similarity of vectors $\mathbf{a}$ and $\mathbf{b}$.

Let the protection cost be

$$c_1(X', S') = \sum_i \min_j \text{los}(\mathbf{s}_i^t - \mathbf{x}_j^a, \mathbf{s}^p - \mathbf{x}_j^a), \quad (27)$$

where $\mathbf{s}_i^t$ is the position of the i th threat, $\mathbf{x}_j^a$ is the position of the j th agent, and $\mathbf{s}^p$ is the position of the protected agent. The protection cost penalizes a formation when agents are not interrupting the line-of-sight from threats to the protected agent. Let the obstacle avoidance (OA) cost be

$$c_2(X', S') = \sum_k \sum_j \|\mathbf{s}_k^o - \mathbf{x}_j^a\|_2^{-1} \quad (28)$$

with $\mathbf{s}_k^o$ as the k th obstacle to avoid. Figure 4 shows the relative locations used to calculate the costs.

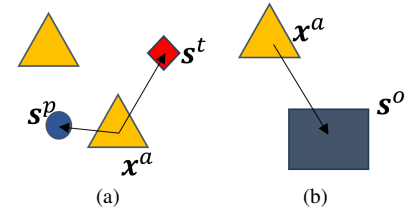


Figure 4: Diagrams showing the agents’ relative locations used for calculating costs. (a) Relative locations for protection cost (27). (b) Relative locations for obstacle cost (28).

The violation cost is a composition of (27) and (28):

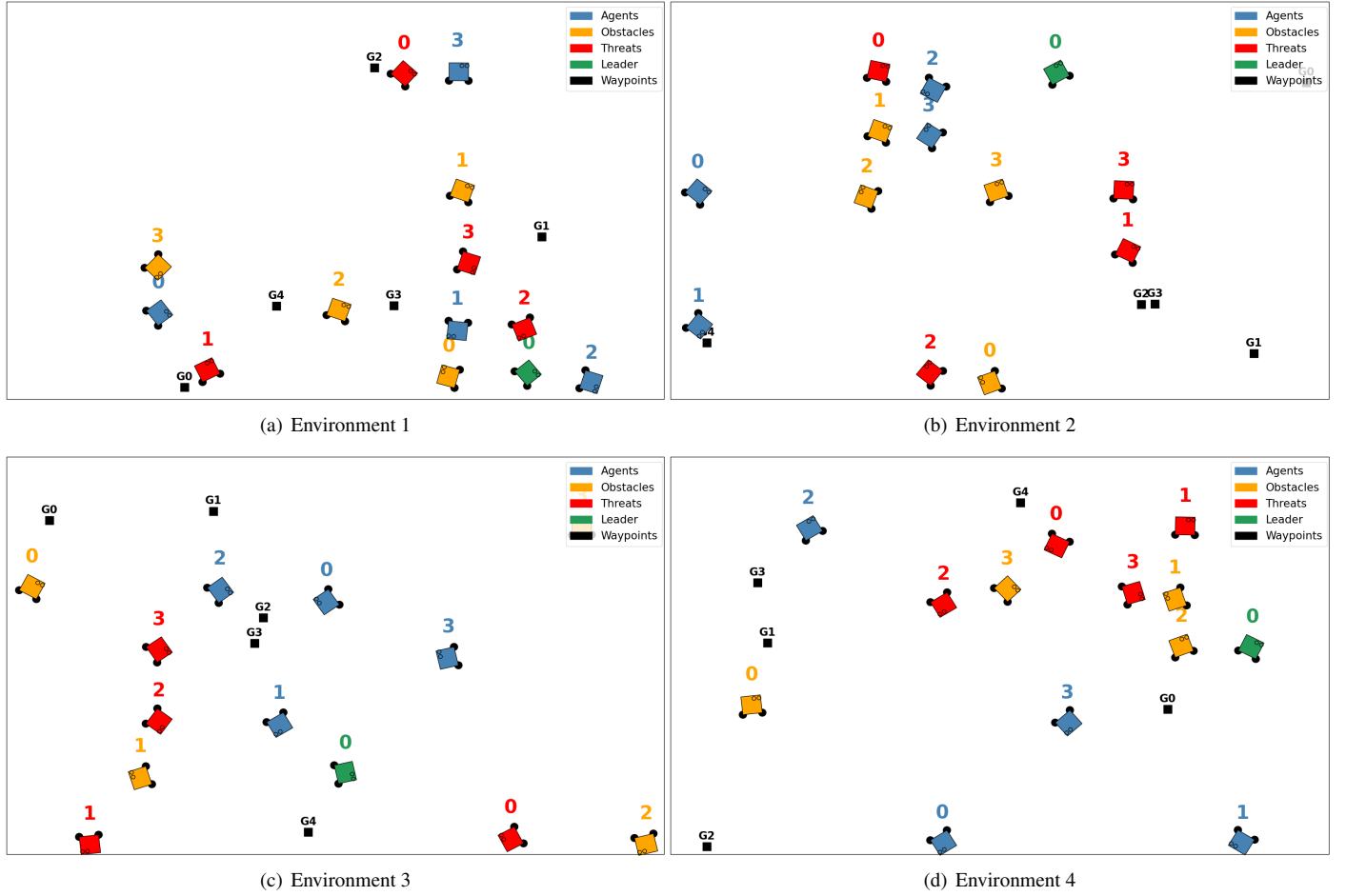


Figure 3: Four different simulated environments ($3.2 \text{ m} \times 2.0 \text{ m}$) for testing. There are four different types of entities in each environment: a leader, agents, obstacles, and threats. Each environment is randomly generated. The black squares indicate the waypoints the leader will follow.

$$c_3(X', S') = \sum_i [\min_j \log(s_i^t - \mathbf{x}_j^a, \mathbf{s}^p - \mathbf{x}_j^a) > \tau_p] + \sum_k \sum_j [\|\mathbf{s}_k^o - \mathbf{x}_j^a\|_2 < \tau_o] \quad (29)$$

where τ_p is the minimum acceptable protection cost for each agent and τ_o is the minimum acceptable distance between obstacles. The brackets $[\]$ denote Iverson's notation as an indicator when the enclosed expression is satisfied (each trigger counts as 1.0). For experiments, $\tau_p = -0.5$ and $\tau_o = 0.2$.

Basis Functions of the Surrogate Cost

Five functions are selected as a basis in the experiment: proximity cost, protection cost (different from (27)), obstacle cost (different from (28)), agent avoidance cost, and payload avoidance cost. These functions quantify the relationships between entities in the environment while maintaining continuity and better-behaved derivatives. Let the proximity cost be

$$f_1(X', S') = \sum_i \|\mathbf{s}^p - \mathbf{x}_i^a\|_2^2, \quad (30)$$

where $\mathbf{x}_i^a$ is the position of the i th agent and $\mathbf{s}^p$ is the position of the payload. The proximity cost quantifies the hazard of agents coming too close to the payload. Let the protection cost be

$$f_2(X', S') = \sum_i \sum_{j \in \mathcal{N}_i} \log(s_j^t - \mathbf{x}_i^a, \mathbf{s}^p - \mathbf{x}_i^a), \quad (31)$$

where $\mathbf{s}_j^t$ is the position of the j th threat. The protection cost penalizes a formation when agents are not interrupting the line-of-sight from threats to the payload. Note this basis function in (31) removes the nonlinear min operation found in the true protection cost (27). Let the obstacle avoidance cost be

$$f_3(X', S') = \sum_i \sum_{k \in \mathcal{N}_i} e^{-\zeta \|\mathbf{s}_k^o - \mathbf{x}_i^a\|_2^2}. \quad (32)$$

In this basis function, we use the exponentiated distance as a similar, better-defined option than the inverse distance used in (28). Let the agent avoidance cost be

$$f_4(X', S') = \sum_i \sum_{i' \in \mathcal{N}_i} e^{-\zeta \|\mathbf{x}_{i'}^a - \mathbf{x}_i^a\|_2^2}. \quad (33)$$

Let the payload avoidance cost be

$$f_5(X', S') = \sum_i e^{-\zeta \|\mathbf{s}^p - \mathbf{x}_i^a\|_2^2}. \quad (34)$$

Note that each cost is an explicit function of a relative quantity. Since formations model physical relationships, a formation is the correct model for this experiment.

Evaluation Metrics

Each of the four formation systems are compared using costs normalized to the least cost among all the techniques in each environment. These relative costs are used instead of raw costs to emphasize the comparison between different techniques. Since the relative costs are unit-less, the relative costs for each technique are summed to create a single metric that describes how well a technique performs overall. This sum is shown in the last row of Tables 2 and 3.

Using Formation Planning for Protection

Relative to static and affine formation shapes, formation planning reduces the protection cost (27) by 75%. Table 2 shows these results with the middle double column dividing formation planning approaches from approaches that use a predetermined shape. This significant reduction in cost comes from the fact that the protection cost assumes the environment is known. Using a static formation or even geometric measures of distance to determine how well a formation achieves the agents' purpose is not sufficient.

Table 2: Relative Protection Cost of Different Formation Systems

		FP-AW		FP		FS		AF
Env 1		1.00		1.22		3.59		5.69
Env 2		1.80		1.00		4.21		11.40
Env 3		1.18		1.00		3.12		7.76
Env 4		1.04		1.00		4.26		4.85
Total Sum		5.02		4.22		15.18		29.71

Implications of Theorem 1

FP-AW performs on par with FP according to Table 2. When the agents are far away from obstacles, or whenever multiple cost functions are nearly flat, Theorem 1 informs that there are multiple sets of weights that will find the same minimum. This is analagous to multiple teams of humans able to reach the same goals through a variety of different means. In situations where Theorem 1 holds, it may be more advantageous to use a set of weights rather than spending time and resources trying to find the perfect set of weights.

Using Formation Planning with Different Costs

The total relative cost of using a formation system with a formation planner can be 20-40% lower than the relative cost of using a formation system without a formation planner. Table 3 shows the relative true cost when using the protection cost (P) from (27), obstacle cost (O) from (28), and violation cost (V) from (29) as true costs for comparison.

Table 3: Relative Cost of Different Formation Systems

	True Cost		FP-AW		FP		FS		AF
Env 1	P		1.00		1.22		3.59		5.69
	O		1.00		16.65		6.20		1.41
	V		1.00		1.12		1.24		1.24
Env 2	P		1.80		1.00		4.21		11.40
	O		3.73		14.52		4.73		1.00
	V		1.07		1.00		1.17		2.03
Env 3	P		1.18		1.00		4.21		11.40
	O		2.29		15.51		2.59		1.00
	V		1.30		1.00		1.01		1.52
Env 4	P		1.04		1.00		4.26		4.85
	O		10.59		27.21		12.75		1.00
	V		1.00		1.75		1.49		1.32
Total Sum			27.00		82.98		46.36		40.23

The FP-AW method has the lowest relative true cost because it is able to change the weights in the surrogate cost function to match the true cost as best as possible. Table 2 shows both formation planning methods perform well, but the FP method performs the worst according to relative costs in Table 3. Adaptive weights make

formation planning more generalizable. To show the effect of adaptive weights, consider a trial from Environment 3 when using the true obstacle avoidance cost analyzed in Figure 5. The FP-AW method is able to adjust the surrogate cost such that a more applicable basis function, payload avoidance, is the primary contribution to $c_t(\mathbf{x})$ instead of the protection basis function.

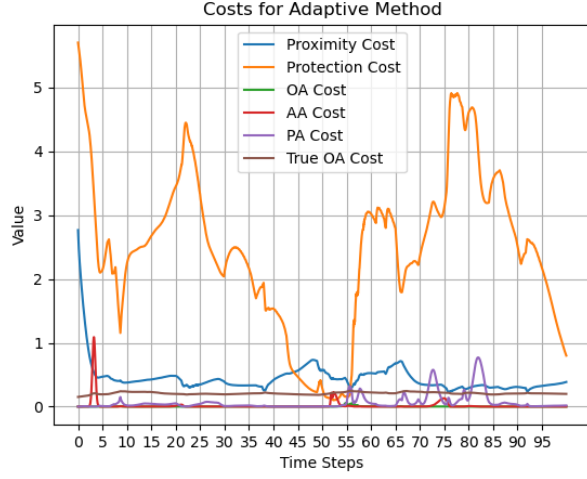
Figure 6 shows an example of the formations created by all 4 methods when using violation cost (29) as the true cost. Figures 6(a) and 6(b) are similar, but Figure 6(a) shows the FP-AW method is able to avoid obstacles better than the FP method in Figure 6(b) while still protecting the leader. The paths taken by agents using the FP-AW method are not as smooth because the formation planner can quickly change which cost functions are the most important. The ability of the other methods to protect the payload is fairly poor, with some agents so far away they cannot be seen in the figure.

Conclusion

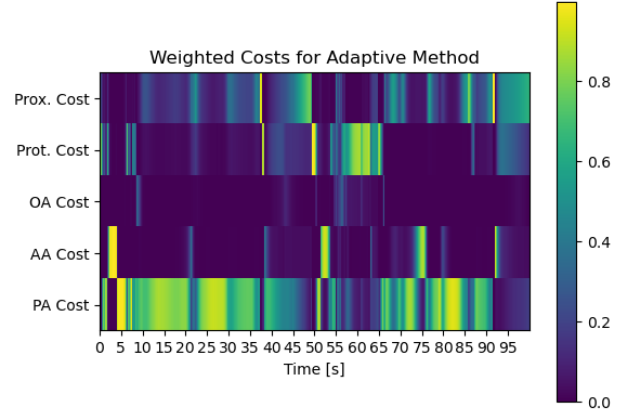
We have shown the inclusion of formation planning to formation systems significantly improves the ability to solve the right problem. When focusing on a single outcome, we saw formation planning have costs 75% lower than formation controller costs. Using an average of considering three distinct true costs, we saw a 20-40% reduction in cost. As exteroceptive sensing continues to improve, formation planning with respect to the environment will enable multi-agent systems to achieve their intended objectives much more effectively than selecting a formation from a finite list. Building on our approach by considering the theoretical limitations of formation planning in Theorem 1 will ensure heterogeneous multi-agent systems complete mission objectives more safely and efficiently.

References

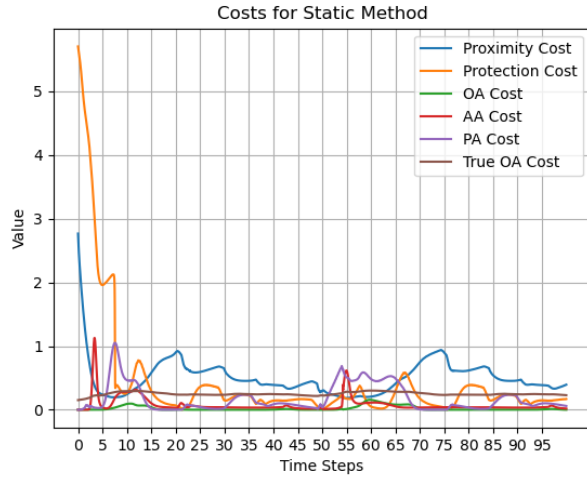
1. E. Salas, D. L. Reyes, and S. H. McDaniel, "The science of teamwork: Progress, reflections, and the road ahead," *American Psychologist*, vol. 73, no. 4, pp. 593–600, May 2018.
2. [Online]. Available: <https://skyelementsdrone.com/>
3. M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, "Three-dimensional multirobot formation control for target enclosing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 357–362. [Online]. Available: <https://ieeexplore.ieee.org/document/6942584>
4. G. López-Nicolás, M. Aranda, and Y. Mezouar, "Adaptive multirobot formation planning to enclose and track a target with motion and visibility constraints," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 142–156, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8877807>
5. V. D. Sharma, L. Zhou, and P. Tokekar, "D2coplan: A differentiable decentralized planner for multi-robot coverage," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3425–3431. [Online]. Available: <https://ieeexplore.ieee.org/document/10160341>
6. A. Bacula, E. Villalovoz, D. Flynn, A. Mehta, and H. Knight, "Social triangles and aggressive lines: Multi-robot formations impact navigation and approach," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 4014–4020.
7. Y. A. Hasan, A. Garg, S. Sugaya, and L. Tapia, "Defensive escort teams for navigation in crowds via multi-agent deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5645–5652, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9143392>
8. Y. Shapira and N. Agmon, "Path planning for optimizing survivability of multi-robot formation in adversarial environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4544–4549. [Online]. Available: <https://ieeexplore.ieee.org/document/7354023>
9. K. Li, J. Wang, and Y. Zheng, "Cooperative formation of autonomous vehicles in mixed traffic flow: Beyond platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 951–15 966, 2022.
10. W. Liu, J. Hu, H. Zhang, M. Y. Wang, and Z. Xiong, "A novel graph-based motion planner of multi-mobile robot systems with formation and obstacle constraints," *IEEE Transactions on Robotics*, vol. 40, pp. 714–728, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10345724>
11. C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: Association for Computing Machinery, 1987, p. 25–34. [Online]. Available: <https://doi.org/10.1145/37401.37406>
12. T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995. [Online]. Available: https://web.mit.edu/~jadbabai/www/ESE680/Vicsek_SPP.pdf
13. J. Leitner, "Formation flying-the future of remote sensing from space," in *18th International Symposium on Space Flight Dynamics*, vol. 548, 2004, p. 621.
14. J. R. T. Lawton, *A behavior-based approach to multiple spacecraft formation flying*. Brigham Young University, 2000.
15. J. Judson, "Us army developing integrated formations of robots and humans," *Defense News*. [Online]. Available: <https://www.defensenews.com/unmanned/2023/10/09/us-army-developing-integrated-formations-of-robots-and-humans/>
16. A. F. F. Silva, L. E. Almeida, and D. G. Macharet, "Efficiently approaching groups of people in a socially acceptable manner in environments with obstacles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 038–12 044.
17. Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, "Necessary and sufficient graphical conditions for affine formation control," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2877–2891, 2016.
18. S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8270608>
19. S. Oughourli, M. Limbu, Z. Hu, X. Wang, X. Xiao, and D. Shishika, "Team coordination on graphs with state-dependent edge cost," 2023. [Online]. Available: <https://arxiv.org/pdf/2303.11457.pdf>
20. C. A. Dimmig, K. C. Wolfe, and J. Moore, "Multi-robot planning on dynamic topological graphs using mixed-integer programming," 2023. [Online]. Available: <https://arxiv.org/pdf/2303.11966.pdf>
21. *FM 3-90 Tactics*, Department of the Army, Washington, D.C., May 2023, available at https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN38160-FM_3-90-000-WEB-1.pdf.
22. J. G. R. III, "Tactical behaviors for autonomous maneuver: collaborative research program (TBAM-CRP)," in *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2023*, R. Suresh, Ed., vol. 12544,



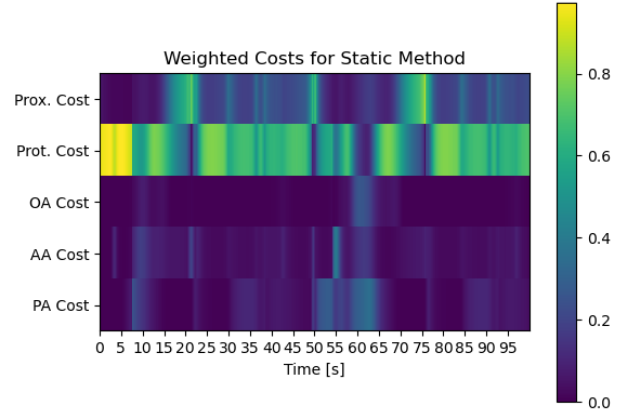
(a) Costs for FP-AW Method



(b) Weighted Costs for FP-AW Method



(c) Costs for FP Method



(d) Weighted Costs for FP Method

Figure 5: Costs for the FP-AW and FP methods in Environment 3. The FP-AW method was able to perform better than the FP method by shifting focus to costs in the surrogate cost that affect the true obstacle avoidance cost. We see this in effect in how the FP-AW is able to achieve a low true obstacle avoidance cost by ignoring the proximity cost.

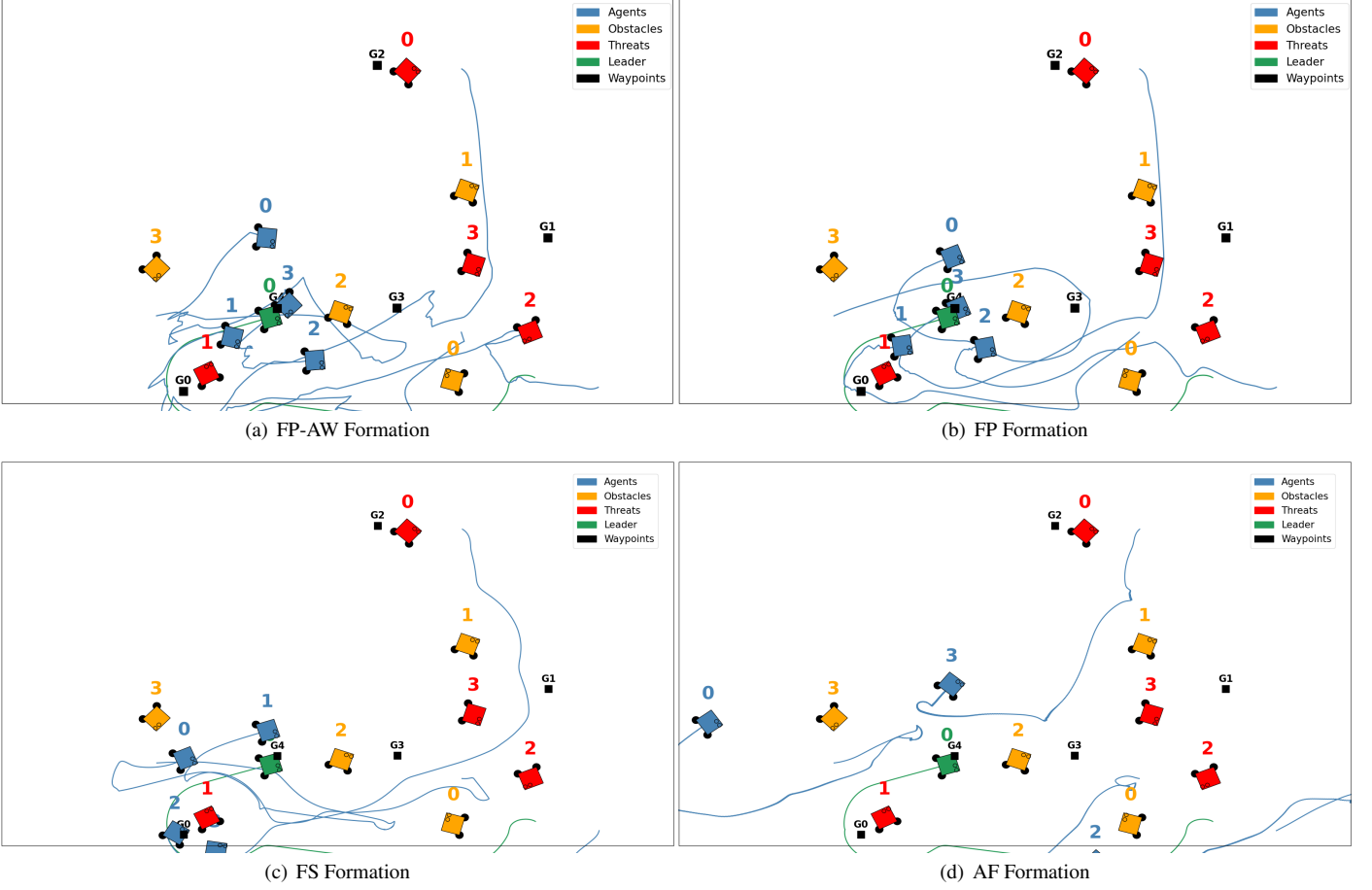


Figure 6: The FP-AW method reduces the violation cost the most because it is able to protect and avoid obstacles. This snapshot is taken at $t = 36$ in Environment 1. The blue and green traces denote the paths taken by the agents and leader, respectively. We see effective obstacle and threat avoidance by the traces that go around these entities. In sub-figures 6(c) and 6(d), we can see the agents are not in a protective position, blocking line-of-sight between the threats and the payload. The FP-AW and FP methods shown in sub-figures 6(a) and 6(b) are more protective. The FP-AW method exhibits slightly better payload avoidance with agent 3 not as near to the payload as in the FP method.

- International Society for Optics and Photonics. SPIE, 2023, p. 125440D. [Online]. Available: <https://doi.org/10.1117/12.2663932>
23. S. Han, S. Yang, Y. Lee, M. Lee, J.-i. Park, and K.-S. Kim, “The leader-follower formation control of nonholonomic vehicle with follower- stabilizing strategy,” *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 707–714, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10224906>
 24. F. Båberg and P. Ögren, “Formation obstacle avoidance using rrt and constraint based programming,” in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2017, pp. 1–6.
 25. T. Balch and R. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/736776>
 26. L. Barnes, M. Fields, and K. Valavanis, “Unmanned ground vehicle swarm formation control using potential fields,” in *2007 Mediterranean Conference on Control and Automation*, 2007, pp. 1–8. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4433724>
 27. G. Antonelli, F. Arrichiello, and S. Chiaverini, “Experiments of formation control with multirobot systems using the null-space-based behavioral control,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1173–1182, 2009. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4814533>
 28. G.-P. Liu and S. Zhang, “A survey on formation control of small satellites,” *Proceedings of the IEEE*, vol. 106, no. 3, pp. 440–457, 2018. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8293791>
 29. Z. Zhang, Y. Li, Z. Gu, and Z. Wang, “Formation rotation and assignment: Avoiding obstacles in multi-robot scenarios,” *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 7015–7022, 2024.
 30. S. Xu, W. Zhang, L. Zhu, and C. P. Ho, “Distributed model predictive formation control with gait synchronization for multiple quadruped robots,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9995–10 002. [Online]. Available: <https://ieeexplore.ieee.org/document/10161260>
 31. A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, “Graph policy gradients for large scale robot control,” in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 823–834. [Online]. Available: <https://proceedings.mlr.press/v100/khan20a.html>
 32. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017. [Online]. Available: <https://arxiv.org/abs/1609.02907>
 33. C. Bai, P. Yan, W. Pan, and J. Guo, “Learning-based multi-robot formation control with obstacle avoidance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 811–11 822, 2022.
 34. J. Branke, *Multiobjective optimization: Interactive and evolutionary approaches*. Springer Science & Business Media, 2008, vol. 5252.
 35. J. Alonso-Mora, E. Montijano, T. Nägele, O. Hilliges, M. Schwager, and D. Rus, “Distributed multi-robot formation control in dynamic environments,” *Autonomous Robots*, vol. 43, pp. 1079–1100, 2019. [Online]. Available: <https://link.springer.com/article/10.1007/s10514-018-9783-9>
 36. J. Nocedal and S. J. Wright, *Numerical optimization: Second edition*. Springer, 2006.
 37. A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, pp. 25–57, 2006.
 38. H. R. G. W. Verstraete, S. Wahls, J. Kalkman, and M. Verhaegen, “Model-based sensor-less wavefront aberration correction in optical coherence tomography,” *Opt. Lett.*, vol. 40, no. 24, pp. 5722–5725, Dec 2015. [Online]. Available: <https://opg.optica.org/ol/abstract.cfm?URI=ol-40-24-5722>
 39. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
 40. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
 41. S. Wilson, P. Glotfelter, S. Mayya, G. Notomista, Y. Emam, X. Cai, and M. Egerstedt, “The robotarium: Automation of a remotely accessible, multi-robot testbed,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2922–2929, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9366341>