A GRADIENT-BASED METHOD FOR TEAM EVASION

Shih-Yuan Liu
Department of Mechanical Engineering
University of California, Berkeley
Berkeley, California, 94720
Email: syliu@berkeley.edu

Zhengyuan Zhou
Department of EECS
University of California, Berkeley
Berkeley, California, 94720
Email: zhengyuan@berkeley.edu

Claire Tomlin
Department of EECS
University of California, Berkeley
Berkeley, California, 94720
Email: tomlin@eecs.berkeley.edu

Karl Hedrick
Department of Mechanical Engineering
University of California, Berkeley
Berkeley, California, 94720
Email: khedrick@me.berkeley.edu

ABSTRACT

We formulate the problem of an autonomous agent team facing the attack of an adversarial agent as a single-pursuer-multiple-evader pursuit-evasion game, with the assumption that the pursuer is faster than all evaders. In this game, the pursuer aims to minimize the capture time of the last surviving evader, while the evaders as a team cooperate to maximize this time. We present a gradient-based approach that quickly computes the controls for the evaders as a team under an open-loop formulation that is conservative towards the evader team by deriving analytical formulas. We demonstrate the advantage of the gradient-based approach by comparing performance both in computation time and in optimality with the iterative open-loop method studied in our previous work. Multiple heuristics have been designed to deal with the inherent intractability of evaluating all possible capture sequences. Extensive simulations have been performed, with results discussed. [1]

INTRODUCTION

In the past decades, autonomous agent teams have received increasing attention due to their wide ranging applications. One interesting application is to deploy a team of autonomous agents, such as unmanned aerial vehicles (UAVs), to cooperatively collect information from a region of interest as in [1]. One key challenge faced by these autonomous agent teams is that the region of interest might be an adversarial environment where hostile agents with conflicting objectives exist, aiming to take down the vehicles in the team. In scenarios where an adversarial agent attempts to attack the team, the problem can be modeled as a single-pursuer-multiple-evader problem. There has been a rich body of literature with various formulations that differ in the capabilities of the agents and the objective functions. In [2], it is assumed that the pursuer has a limited detection range and that the two evaders as a team cooperate to maximize the minimum of the two corresponding distances to the pursuer. [3] follows a similar formulation and [4] solves the problem in a 3-dimensional space. In [5], a special objective function is designed to encourage the evaders to be on the opposite side of the pursuer. [6] provides a good review on various different objective functions that are commonly utilized in the single-pursuer-multiple-evader problem.

In this paper we consider a pursuit-evasion game in a 2-dimensional space with the objective function being the survival time of the last captured evader. We assume it to be a full-knowledge pursuit-evasion game in that each agent knows the exact locations of all other agents at all times. In [7], one of the seminal works in this area, Breakwell *et al.* have shown that for some initial conditions the optimal strategy for the evaders are to travel in straight lines with maximum speed, and the cor-

[1]This work has been supported by ONR under the HUNT MURI (N0014-08-0696).

responding optimal trajectory of the pursuer are line segments connecting the successive capture points of the evaders. In other words, for these initial conditions, neither the optimal capture sequence nor the optimal headings change along the optimal trajectories of the pursuer and the evaders. However, for other initial conditions, this result does not hold. The optimal strategy has to be computed by backwards integration of the Hamilton-Jacobi-Isaacs (HJI) equation and is hardly generalizable to cases with more than two evaders, as the state space scales exponentially with the number of players, rendering the computations infeasible for multi-evader scenarios. Chikrii *et al.* investigated a slightly different scenario where the pursuer must capture the evaders in a specific order. It has been shown in [8] that the optimal strategy for the evaders in this fixed-sequence formulation is to travel with constant heading and maximum speed for all initial conditions. Therefore, the fixed-sequence problem can be formulated as an directional optimization problem with the evaders' joint heading being the optimization variable. This problem is solved numerically in [9] for cases with more than two evaders. Recently Belousov *et al.* have in [10] converted the fixed-sequence problem from a N-dimensional unconstrained non-linear optimization problem to a one-dimensional root searching problem by exploiting the first order optimality condition. As a result, the optimal joint headings of the evaders can be solved very efficiently even for a large number of evaders. In [11] and [12], the fixed-sequence problem with constraints on the pursuer's turning rate is solved.

The fixed-sequence formulation provides a conservative mind-set for the pursuer in that the pursuer must pick a capture sequence first and later commit to it, while the team of evaders then cooperatively optimize against the purser given the knowledge of such a capture sequence already chosen by the pursuer. In [8], it has been shown that the optimal capture sequence of the fixed-sequence problem depends not only on the relative positions of the agents but also on the speed ratios. In [13], Berdyshev utilized a space partitioning technique from the Traveling Salesman Problem (TSP) literature to solve for the optimal capture sequence more efficiently. As pointed out by Serov in [14], the fixed-sequence problem can be viewed as a special case of the Generalized Traveling Salesman Problem (GTSP) where each city can pick its location from a given set of locations after the salesman has declared its traveling route. Both problems (TSP and GTSP) are known to be NP-complete. Yet, heuristics can be designed to speed up the computations at some expense of optimality.

While the optimal evader actions can be efficiently computed in a fixed-sequence formulation, in a more realistic pursuit-evasion scenario the pursuer is not likely to be enforced to capture the evaders according to a specific sequence. We call this formulation where the pursuer is not obligated to capture the evaders in a specific sequence the sequence-free formulation. It has been shown in [7] that in the 2-evader case, for initial conditions that satisfy some specific geometrical conditions the optimal trajectories of the fixed-sequence formulation are the optimal trajectories for the sequence-free formulation. However, with more than 2 evaders there is not a way to identify these initial conditions other than actually simulating the whole chase. Moreover, it is not clear how the evaders should act under these initial conditions in that the backwards integration approach in [7] requires some geometrical properties that are only true in the 2-evader case. The evaders' controls in the sequence-free formulation are somewhat incomplete in the literature. To address those issues, we have in [15] proposed an open-loop formulation of such pursuit-evasion games. There, we have provided a conservative strategy for the evaders as a team that is valid for all initial conditions and that provides guaranteed survival time. Moreover, we have proposed an iterative open-loop scheme to relax the conservatism, hence achieving better performance from the evaders' point of view. However, the iterative open-loop scheme is computationally expensive due to the fact that an optimization problem has to be solved at every time step. Hence, in this paper, we propose a more efficient gradient-based method which bypasses the need of solving the optimization problem and can achieve similar survival time performance for the evaders.

PROBLEM FORMULATION

Consider a pursuit-evasion game taking place in $\mathbb{R}^2$, where N evaders attempt to evade from a single pursuer that is faster than all evaders. Without loss of generality, we assume that the pursuer's maximum speed is 1 and the i-th evader's maximum speed is β_i, $0 \leq \beta_i < 1$, $i = 1, \ldots, N$. The pursuer aims to capture all the evaders as soon as possible, while the evaders, as a team, attempt to delay the capture time of the last evader for as long as possible. An evader is considered captured if at some point, the pursuer coincides with the evader. The point capture can always be achieved in finite time since the pursuer is strictly faster than all the evaders. The control for agent j is $u_j(\cdot) : \mathbb{R}^+ \to \mathbb{R}^2$, where the subscript $j = p$ denotes the pursuer and the subscript $j = i$ ($1 \leq i \leq N$) denotes evader i. The trajectory of agent j is $x_j(\cdot) : \mathbb{R}^+ \to \mathbb{R}^2$. We assume the following simple motion dynamics, where $\mathbf{x}(\cdot)$ is the joint trajectory of all the players:

$$\dot{x}_p(t) = u_p(t), \tag{1}$$

$$\dot{x}_i(t) = \beta_i u_i(t), \qquad i = 1, \ldots, N \tag{2}$$

$$\mathbf{x}(0) = \mathbf{x}^0. \tag{3}$$

It is understood that each u_j is an element of the set $\mathbf{U}$ of admissible controls: where $\mathbf{U} = \{u_j(\cdot) | \|u_j(t)\| \leq 1, t \in [0, \infty)\}$ In other words, u_j represents the direction controlled by agent j as well as the ratio of agent j's speed over its maximum speed. We also note that the control set $\mathbf{U}$ is independent of which j we choose and that $\mathbf{U}^N$ denotes the joint admissible control set of the N evaders. For convenience, we define $\mathbf{u}_e(\cdot) = [u_1(\cdot), \ldots, u_N(\cdot)]$ to be the joint control of all the evaders. Define $\mathbf{S}_N = \{[s_1, \ldots, s_N] | s_i \in \mathbb{N}_{>0}, s_i \neq s_j \text{ for } i \neq j, \max_i s_i = N\}$ to be the set of all permutations of the sequence 1 through N, representing all possible sequences that the evaders can be captured.

Definition 1. *For a given initial condition $\mathbf{x}(0)$, a fixed pursuer's control $u_p(\cdot)$ and a fixed joint control $\mathbf{u}_e(\cdot)$ of all the evaders, we define the capture time of the i-th evader to be:* $\tau_i(\mathbf{x}(0), u_p(\cdot), \mathbf{u}_e(\cdot)) = \inf\{t | x_p(t) = x_i(t)\}$.

In other words, τ_i is the time it takes for the pursuer to capture evader i if each agent adopts the prescribed control respectively under the joint initial condition. If evader i is not captured with the prescribed control, the value of τ_i is defined to be infinity. For cases where all evaders are captured eventually we have $x_i(\tau_i) = x_p(\tau_i)$, $i = 1, \ldots, N$. Note that the total time of capturing all the evaders can be represented by $\sup_{i\in\{1,\ldots,N\}} \tau_i$.

In [15], we proposed an open-loop formulation of the problem to address the curse of dimensionality in the classical closed-loop, Hamilton-Jacobi-Isaacs (HJI) formulation. The open-loop formulation is characterized by the following two definitions.

Definition 2. *For a given initial condition $\mathbf{x}^0$ and a given joint control $\mathbf{u}_e(\cdot)$ for all the evaders, the minimum capture time of the team is defined to be*

$$T^\star(\mathbf{x}^0, \mathbf{u}_e(\cdot)) = \inf_{u_p(\cdot)\in\mathbf{U}} \sup_{i\in\{1,\ldots,N\}} \tau_i(\mathbf{x}^0, u_p(\cdot), \mathbf{u}_e(\cdot)). \quad (4)$$

That is, $T^\star$ is the minimum time it takes for the pursuer to capture all the evaders given that the pursuer knows the prescribed control functions of all the evaders.

Definition 3. *For a given initial condition $\mathbf{x}^0$, the optimal open-loop capture time of the game is defined to be*

$$\tau^{ol\star}(\mathbf{x}^0) = \sup_{\mathbf{u}_e(\cdot)\in\mathbf{U}^N} T^\star(\mathbf{x}^0, \mathbf{u}_e(\cdot)). \quad (5)$$

We note that $\tau^{ol\star}$ essentially encodes the conservatism towards the evaders. The information pattern embodied in the definition is that the evaders first jointly choose the optimal control functions over the entire time horizon assuming the worst case. Thereafter, all the evaders must commit to the control functions chosen initially. Then, with the chosen joint control of all the evaders revealed to the pursuer, the pursuer then selects its best control to minimize the total capture time.

We have shown in [15] that the evaders should travel in constant headings with maximum speed for optimality. Hence the control of evader i is fully specified by its heading θ_i. The optimal action for the pursuer with a given capture sequence is to continue capturing the next evader in the sequence using the parallel pursuit strategy given in [10]. Let $\mathbf{x}$ be an arbitrary joint state of N evaders and one pursuer, θ be the joint heading of the evaders, $T_s^\star(\mathbf{x}, \theta)$ denotes the minimum time it takes for the pursuer to capture all the evaders according to the specific capture sequence $s \in \mathbf{S}_N$. The open-loop survival time $\tau^{ol\star}(\mathbf{x})$ and the optimal joint heading $\theta^{ol\star}(\mathbf{x})$ of the evaders can be expressed respectively as follows:

$$\tau^{ol\star}(\mathbf{x}) = \sup_{\theta} \inf_{s\in\mathbf{S}_N} T_s^\star(\mathbf{x}, \theta), \quad (6)$$

$$\theta^{ol\star}(\mathbf{x}) = \operatorname{argsup}_{\theta} \inf_{s\in\mathbf{S}_N} T_s^\star(\mathbf{x}, \theta). \quad (7)$$

In the sup-inf problem defined in Eq. (6), for a given $\mathbf{x}$ the sup player is maximizing over the point-wise minimum of a finite set of smooth functions $\{T_s^\star(\mathbf{x}, \cdot)\}_{s\in\mathbf{S}_N}$. This problem can be re-formulated as a constrained nonlinear optimization problem using the procedures outlined in [16, Chap. 2]. In this work we use the MATLAB function `fminimax`, which utilizes sequential quadratic programming (SQP) in [17], to solve the sup-inf problem.

While our open-loop approach provides worst-case guarantees to the performance of the evaders as a team, it tends to be conservative. Hence, we have in [15] also developed an iterative open-loop scheme where the evaders update their headings every Δt seconds according to the optimal open-loop joint heading of the most up-to-date joint state. Note that at each update time the evaders are still solving an open-loop problem where they assume they cannot change their headings afterwards. This scheme relaxes the conservatism inherent in the open-loop formulation and improves the survival time performance of the evaders.

While the iterative open-loop scheme provides a promising venue by relaxing the conservatism, it is also computationally expensive since at every iteration a constrained non-linear optimization problem must be solved. Moreover, it entails centralized computations and cannot leverage the distributed computing power. In this paper, we propose a gradient-based method that can achieve similar survival time performance as the iterative open-loop scheme with much less computing power. Moreover, this gradient-based method can exploit the distributed computing power available and further decrease the required computation time.

THE GRADIENT-BASED METHOD

According to the problem formulation in Eq. (7), the evaders are solving for the maximizer of the function $\inf_{s\in\mathbf{S}_N} T_s^\star(\mathbf{x}, \theta)$. Although the function $T_s^\star(\mathbf{x}, \theta)$ is C^1 continuous for any specific capture sequence s [15], the point-wise minimum of all these functions is not. Hence we can not apply a gradient accent method by computing the gradient of $\inf_{s\in\mathbf{S}_N} T_s^\star(\mathbf{x}, \theta)$ with respect to θ directly. Here we propose a gradient-based method which exploit the fact that $\nabla_\theta T_s^\star(\mathbf{x}, \theta)$ is well defined for all $\mathbf{x}$ and θ for any specific capture sequence s.

Given that the evaders are able to update their headings every Δt seconds, the joint heading should be updated according to the following formula:

$$\theta' = \theta + c_\theta \Delta t \frac{\nabla_\theta T^\star_{s^\star(\mathbf{x},\theta)}(\mathbf{x},\theta)}{\|\nabla_\theta T^\star_{s^\star(\mathbf{x},\theta)}(\mathbf{x},\theta)\|_\infty}, \tag{8}$$

$$s^\star(\mathbf{x},\theta) = \underset{s\in\mathbf{S}_N}{\operatorname{arginf}}\, T^\star_s(\mathbf{x},\theta). \tag{9}$$

The joint heading is updated toward the direction of the gradient of the minimum capture time of the optimal capture sequence with respect to the current joint heading. The step size is determined by the parameter c_θ and the update time step Δt. Similar to the iterative open-loop scheme, the evaders will keep this updated joint heading θ' for Δt seconds and then repeat the process at the next update time. The algorithm used to compute the key elements of this method, $\nabla_\theta T^\star_{s^\star(\mathbf{x},\theta)}(\mathbf{x},\theta)$ and $s^\star(\mathbf{x},\theta)$, will be described in the following two subsections respectively.

Computing the Gradient given a Capture Sequence

Without loss of generality, we assume that we are computing $\nabla_\theta T^\star_s(\mathbf{x},\theta)$ for a specific capture sequence $s = \{1,\ldots,N\}$. That is, the evaders are captured according to their indices. The gradient of a different capture sequence can be computed using the same procedure simply by re-indexing the evaders.

We start by defining the following three notations:

$$\hat{\tau}_i = \tau_i - \tau_{i-1}, \tag{10}$$

$$\hat{x}_i = x_i(\tau_{i-1}) - x_p(\tau_{i-1}), \tag{11}$$

$$e_{\theta_i} = [\cos\theta_i, \sin\theta_i]. \tag{12}$$

Here $\hat{\tau}_i$ is the difference between the survival time of the i-th and the $(i-1)$-th evader. $\hat{x}_i$ denotes the relative position from the pursuer to the i-th evader when the $(i-1)$-th evader is captured. e_{θ_i} is the unit vector indicating the heading direction of the i-th evader.

At time $t = \tau_{i-1}$, the i-th evader is positioned at $\hat{x}_i$ relative to the pursuer and is traveling in a straight line with heading θ_i and speed β_i. The minimum time it takes for the pursuer to capture the i-th evader can be computed by the following formula:

$$\hat{\tau}_i = \frac{\beta_i}{1^2 - \beta_i^2}\left((\hat{x}_i\cdot e_{\theta_i}) + \sqrt{(\hat{x}_i\cdot e_{\theta_i})^2 + (\frac{1}{\beta_i^2} - 1)(\hat{x}_i\cdot\hat{x}_i)}\right). \tag{13}$$

Note that $\hat{\tau}_i$ depends on $\hat{x}_i$ which in turns depends on $\hat{\tau}_{i-1}$. By setting $\hat{\tau}_0 = 0$, we can compute the value of $\hat{\tau}_i$ sequentially with increasing i. With this procedure, we can compute the time it takes for the pursuer to capture the team of evaders according to a specific capture sequence using the parallel pursuit strategy. The computation scales linearly with the number of evaders. As pointed out in [15], the optimal strategy for the pursuer in this scenario is to capture each evader as soon as possible according to the capture sequence. Hence, the optimal survival time of the last evader in a specific capture sequence s can be written as the sum of all the $\hat{\tau}_i$.

$$T^\star_s(\mathbf{x},\theta) = \sum_{j=1}^{N}\hat{\tau}_j. \tag{14}$$

By definition the gradient of $T^\star_s(\mathbf{x},\theta)$ is

$$\nabla_\theta T^\star_s(\mathbf{x},\theta) = [\sum_{j=1}^{N}\frac{\partial\hat{\tau}_j}{\partial\theta_1}, \sum_{j=1}^{N}\frac{\partial\hat{\tau}_j}{\partial\theta_2},\ldots,\sum_{j=1}^{N}\frac{\partial\hat{\tau}_j}{\partial\theta_N}]. \tag{15}$$

The value of $\frac{\partial\hat{\tau}_j}{\partial\theta_i}$ depends on i and j and can be derived as follows.

$\frac{\partial\hat{\tau}_j}{\partial\theta_i} = 0$ for $j < i$. This is because the heading of an evader does not affect the survival time of the evaders that are captured before it. For $j \geq i$, we have:

$$\frac{\partial\hat{\tau}_j}{\partial\theta_i} = \frac{\beta_j}{1-\beta_j^2}\{A_{i,j} + \frac{1}{B_j}[(\hat{x}_j\cdot e_{\theta_j})A_{i,j} + (\frac{1}{\beta_j^2} - 1)(\hat{x}_j\cdot\frac{\partial\hat{x}_j}{\partial\theta_i})\}, \tag{16}$$

$$B_j = \sqrt{(\hat{x}_j\cdot e_{\theta_j})^2 + (\frac{1}{\beta_j^2} - 1)(\hat{x}_j\cdot\hat{x}_j)}. \tag{17}$$

The terms $A_{i,j}$ and $\frac{\partial\hat{x}_j}{\partial\theta_i}$ depend on the relationship between i and j.

For $j = i$,

$$A_{i,j} = \hat{x}_j\cdot\frac{\partial e_{\theta_j}}{\partial\theta_j}, \tag{18}$$

$$\frac{\partial\hat{x}_j}{\partial\theta_i} = (\beta_j\sum_{k=1}^{j-1}\hat{\tau}_k)\frac{\partial e_{\theta_j}}{\partial\theta_j}. \tag{19}$$

For $j = i+1$,

$$A_{i,j} = \frac{\partial\hat{x}_j}{\partial\theta_i}\cdot e_{\theta_j}, \tag{20}$$

$$\frac{\partial\hat{x}_j}{\partial\theta_i} = \frac{\partial\hat{\tau}_{j-1}}{\partial\theta_i}(\beta_j e_{\theta_j} - \beta_{j-1}e_{\theta_{j-1}}) - \beta_{j-1}\sum_{k=1}^{j-1}\hat{\tau}_k\frac{\partial e_{\theta_{j-1}}}{\partial\theta_{j-1}}. \tag{21}$$

For $j > i+1$,

$$A_{i,j} = \frac{\partial\hat{x}_j}{\partial\theta_i}\cdot e_{\theta_j}, \tag{22}$$

$$\frac{\partial\hat{x}_j}{\partial\theta_i} = (\beta_j e_{\theta_j} - \beta_{j-1}e_{\theta_{j-1}})\sum_{k=1}^{j-1}\frac{\partial\hat{\tau}_k}{\partial\theta_i}. \tag{23}$$

From these equations we can see that the value of $\frac{\partial \hat{\tau}_n}{\partial \theta_m}$ depends on the value of $\hat{\tau}_i$, $i = 1, \ldots, m$ and the values of all $\frac{\partial \hat{\tau}_j}{\partial \theta_i}$ such that $i < m$, $j < n$. Starting from $i = 1, j = 1$, one can compute the value of $\frac{\partial \hat{\tau}_j}{\partial \theta_i}$ by first computing the terms with increasing order of j for $i = 1$, and then repeating the process for $i = 2, 3, \ldots, N$. The number of terms to compute is $\frac{1}{2}N(N+1)$ where N is the number of evaders. Using this procedure, the value of $\nabla_\theta T_s^\star(\mathbf{x}, \theta)$ for a specific $\mathbf{x}$, θ, and s can be computed in polynomial time with respect to the number of evaders.

Determining the Optimal Capture Sequence

In the open-loop formulation, the evaders assume that the pursuer will capture them with the optimal capture sequence which achieves the minimum capture time. When the evaders are stationary, solving for the optimal capture sequence is the same as solving the Traveling Salesman Problem (TSP). For stationary evaders, the distance between them are independent of time and hence the time it takes for the pursuer to capture say evader B right after evader A is always a constant. The optimal capture sequence is only a function of the initial layout $\mathbf{x}$. However, in the open-loop formulation the evaders are not stationary. The time it takes for the pursuer to capture evader B right after evader A not only depends on the initial position of evader A and B, but also the headings of A and B. As pointed out in [14], this additional level of complexity makes this problem more similar to the Generalized Traveling Salesman Problem (GTSP), which like the TSP is also NP-complete.

The optimal sequence can be defined as a function of the layout $\mathbf{x}$ and the joint heading of evaders θ as in Eq. (9). The task of determining such capture sequence $s^\star$ can be viewed as a search problem on a tree. First consider $(\mathbf{x}, \theta)$ with N evaders as the initial state. From this initial state, there are N descendant states each representing the resulting layout when one of the N evaders is captured. Each of these descendant states has a transition cost that is the time it takes for the pursuer to capture the first evader. From each of these descendant states, there are then $N-1$ descendant states based on which of the $N-1$ evaders is captured. The tree grows in this fashion factorially. There are $N!$ final states where all the evaders are captured, and each of the final states represents one possible capture sequence. To compute the accumulated transition cost in a final state, the whole path connecting the initial state to the final state must be traversed and evaluated.

All-Sequence (All-Seq) The brute-force way to find the optimal capture sequence is to evaluate the minimum capture time of all the possible capture sequences. This process can be parallelized by splitting the capture sequences into N groups of $(N-1)!$ capture sequences according to the first evader in the capture sequence. Each evader is then only in charge of evaluating the capture time of $(N-1)!$ capture sequences. A randomly assigned evader can then compare the results, each from one of the N groups and determine the optimal capture sequence. With this parallelism the computation time is proportional to $(N-1)!$ instead of $N!$. However, as the number of evaders increases, it will quickly become computationally intractable.

For computational tractability with a large team of evaders, we employ some pruning techniques with heuristics while traversing the tree so that only a subset of the states are evaluated. For each state we use the heuristics to pick at most K of the possible descendant states to further expand. Here K is the branching factor of the heuristic and there will be at most N^K capture sequences evaluated. Among these N^K captures sequences, the one with the minimum capture time will be returned as the approximate optimal capture sequence. Here we present three different heuristics for picking the K descendant states.

Stepwise-Closest-K (SC-K) This heuristic compares the distances from the evaders that are still not captured to the pursuer, and picks the K closest evaders. Note that when $K = 1$, this heuristic results in a step-wise greedy capture sequence where the evader that is closest to the pursuer is always the next to be captured.

Stepwise-Soonest-K (SS-K) This heuristic compares the minimum time it takes for the pursuer to capture each evader that is still alive, and picks the K soonest-to-capture evaders. Compared to the SC-K, which only makes use of the positions of the evaders, this heuristic makes use of both the positions and the headings of the evaders.

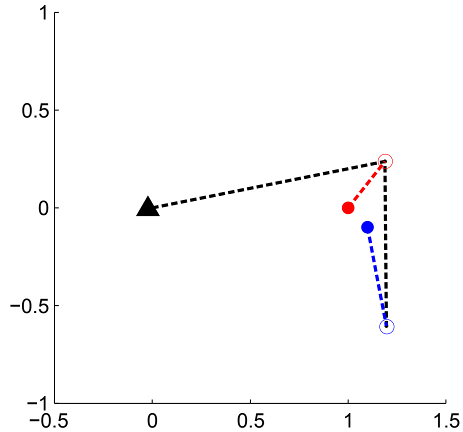
Stepwise-Min-Sum-of-Distance-K (SSD-K) This heuristic compares the sum of distances from the pursuer to all still alive evaders when the pursuer captures a specific evader. It picks the K evaders that will result in a smaller sum of distances when being captured. This heuristic takes into account the positions and headings of the current state and also the positions at the next state.

SIMULATIONS & RESULTS

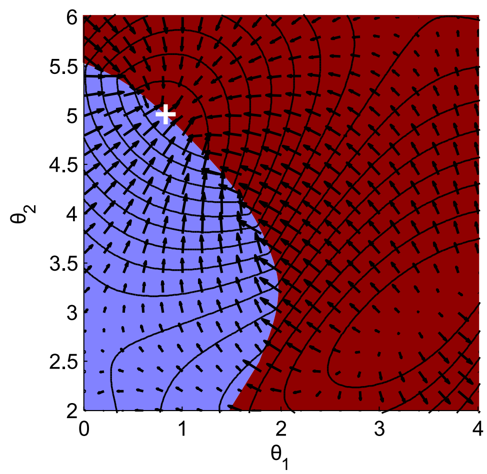
The gradient-based method with various heuristics for capture sequence are implemented and simulated in MATLAB. The details of the simulations and the resulting behavior of the team of evaders are presented in this section. Also, the survival time and computation time performance are presented and compared to that of the iterative open-loop approach proposed in [15].

The Gradient Vector

Figure 1(a) shows a typical 2-evader layout with its optimal open-loop headings and the resulting minimum time capture trajectories. The triangle and the solid circles are the initial position of the pursuer and the evaders respectively. The black, red, and blue dashed lines are the optimal open-loop trajectories of the pursuer and the 2 evaders. The contours in Fig. 1(b) are the level sets of the function $\inf_{s \in \mathbf{S}_N} T_s(\mathbf{x}, \cdot)$. The white cross denotes

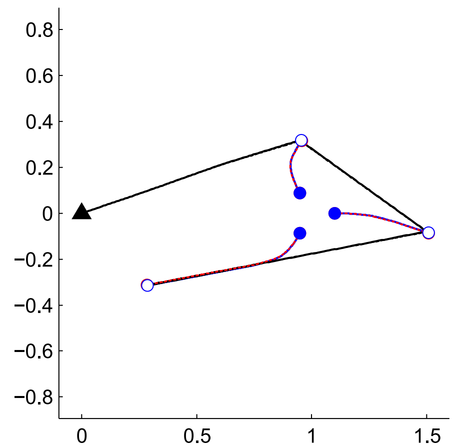


(a) The layout and optimal open-loop trajectories

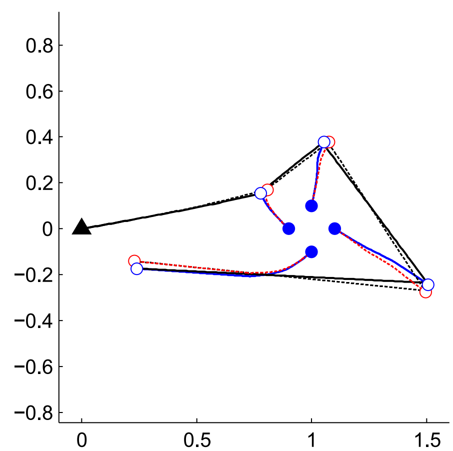


(b) The gradient vector for different joint headings

Figure 1. THE OPTIMAL OPEN-LOOP HEADING AND THE GRADIENT VECTOR OF A 2-EVADER LAYOUT.



(a) 3-evader case



(b) 4-evader case

Figure 2. THE RESULTING TRAJECTORIES OF THE ITERATIVE OPEN-LOOP APPROACH AND THE GRADIENT-BASED APPROACH

the maximizer $\theta^{ol\star}$, which is the optimal joint heading shown in Fig. 1(a). The arrows represent the direction and the magnitude of the gradient vector $\nabla_\theta \inf_{s\in\mathbf{S}_N} T_s(\mathbf{x},\theta)$. The colors of the regions indicate the optimal capture sequence of a given θ. In this case the optimal joint heading $\theta^{ol\star}$ is on the boundary of two regions. This implies that given the optimal joint heading of the evaders the pursuer can achieve the same minimum capture time using either one of the two capture sequences. Note that this is only true under the open-loop scheme where the evaders do not change their joint heading after committing to it at the beginning of the game.

As shown in the figure, the gradient vectors behave nicely near the optimal joint heading as they point toward the optimal joint heading. Also, near the boundaries of the regions the gradient vectors generally point outward from the region where the current θ is in. This implies that by following the gradient vectors, the optimal capture sequence will constantly switch back and forth between the two sequences. With a small enough update time, the minimum capture time of both capture sequences will be essentially kept the same. This behavior is the same as the iterative open-loop approach shown in [15].

Simulations and Resulting Trajectories

The simulations are carried out with an update time $\Delta t = 0.01$sec. The speed of the pursuer is 1 unit per second and the speed of the evaders are set to be 0.25 units per second. The step size parameter c_θ is set to be 2π which means that the evaders can make a 180 degree turn in 0.5sec. There are no limits on the turning rate of the pursuer. At the beginning of every time

Table 1. AVERAGE SURVIVAL AND COMPUTATION TIME OF THE ITERATIVE OPEN-LOOP APPROACH

# of Evaders	2	3	4	5
Survival Time	1.2931s	2.2583s	3.4696s	5.1156s
Computation Time	0.0418s	0.0463s	0.0814s	0.4605s

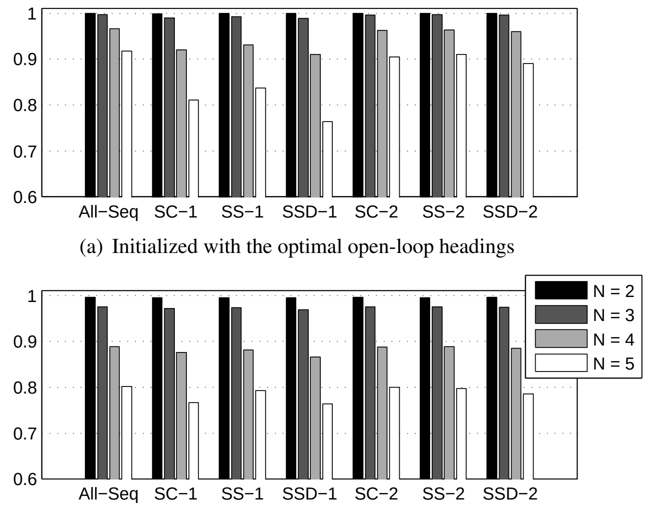
step the iterative open-loop method updates the joint heading of the evaders to the optimal open-loop joint heading $\theta^{ol\star}$ defined in Eq. (6). The gradient-based method updates the joint heading of the evaders according to Eq. (8). After the evaders have updated their headings, the pursuer determines the optimal capture sequence $s^\star(\mathbf{x},\theta)$ by assuming that the evaders will stick to the updated headings until the end of the game. The pursuer then sets its heading to the parallel-pursuit heading for capturing the first evader in the optimal capture sequence with minimum time. For cases where there are multiple optimal capture sequences, the pursuer prefers the one that requires a smaller change in its heading. This additional mechanism prevents the pursuer from switching back and forth between different capture sequences with similar or identical capture time. An evader is considered captured when its distance to the pursuer is closer than or equal to 0.01 units. The simulation terminates when all evaders are captured.

As stated in [15], the pursuer strategy described here is optimal only if the evaders do not change their headings after the game starts. It is sub-optimal against a team of evaders using the iterative open-loop method or the gradient-based method since the evaders' joint heading is updated at every time step. To compute the optimal strategy against these iterative methods, the pursuer needs to solve an optimal control problem where the final state in which all evaders are captured cannot be determined without simulation. Also, the computation grows exponentially with the number of the time steps. Whether or not there exists a computationally tractable method for computing the optimal pursuer strategy against iterative open-loop and gradient-based evaders remains an open problem.

Figure 2 shows the resulting trajectory of the gradient-based approach and the iterative open-loop approach of 3-evader and 4-evader layouts. The solid blue lines and the dashed red lines represent the trajectories of the gradient-based and the iterative open-loop method respectively. In the 3-evader layout the resulting trajectories from the two approaches are almost identical. In the 4-evader layout, the trajectory of the gradient-based approach deviates only slightly from that of the iterative open-loop approach and achieves a slightly lower survival time.

Performance of the Gradient-based Method

To measure the performance of the gradient-based method, 400 random initial layouts are generated for each case of $N = 2,3,4,5$. The pursuer always starts from the origin while the initial positions of the evaders are sampled randomly from a unit square centered at the origin. Simulations are run for each layout using the iterative open-loop approach and the gradient-based approach with various sequence heuristics.



(a) Initialized with the optimal open-loop headings

(b) Initialized directly away from the pursuer

Figure 3. AVERAGE SURVIVAL TIME RATIO OF THE GRADIENT-BASED METHOD TO THAT OF THE ITERATIVE OPEN-LOOP METHOD

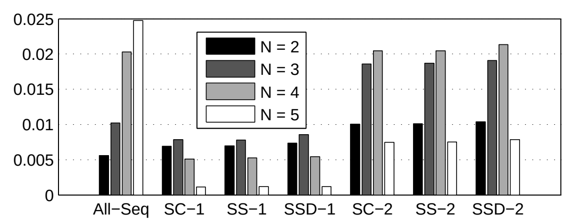


Figure 4. AVERAGE COMPUTATION TIME PER STEP COMPARED TO THAT OF THE ITERATIVE OPEN-LOOP METHOD

We use the performance of the iterative open-loop approach as a baseline to benchmark the performance of the gradient-based approach with different heuristics. Table 1 shows the average team survival time and the average computation time per simulation step for cases with different number of evaders using the iterative open-loop approach. The simulations are run on a laptop with Intel Core i7 dual core CPU M620 at 2.67GHz and 4GB of RAM. The codes are implemented in MATLAB.

Survival Time Performance Figure 3(a) shows the average survival time ratio of the gradient-based method with different sequence heuristics to that of the iteration open-loop method when the evaders are initialized with the optimal open-loop joint heading $\theta^{ol\star}$. For the 2-evader case, the gradient-based method performs almost the same as the iterative open-loop method. The performance degrades slightly as the number of evaders increases. The All-Seq method, which is to evaluate all possible capture sequences to find $s^\star$, achieves about 90% of the survival time achieved by the iterative open-loop method with

5 evaders. As of the performance of the gradient-based method using sequence heuristics with a branching factor of 1, none of the heuristics performs as well as the brute-force method. However, with a branching factor of 2, all three heuristics perform very similarly to the brute-force approach.

The gradient-based method performs almost as well as the iterative open-loop approach when the evaders are initialized with the optimal open-loop joint heading. However to compute this optimal joint heading for the initial layout, one needs to solve the optimization problem in Eq. (7). Although only having to solve it once at the beginning of the game already provides a substantial improvement in the computation time over the iterative open-loop approach, with more evaders it will take a significant amount of computation time. Hence, we designed a heuristic for the initial headings so that the evaders do not have to solve the optimization problem at all. The heuristic is to initialize the evaders' joint heading such that each evader is heading directly away from the pursuer.

Figure 3(b) shows the survival time performance of the gradient-based method with the evaders initialized with the away-from-pursuer headings. The performance degrades slightly compared to the optimally initialized case. The trend of performance degradation with increasing number of evaders is still presented. However, the average survival time is still within 20% of that of the iterative open-loop approach even with a team of 5 evaders. Interestingly, there is not a significant difference in performance between the three different heuristics.

Computation Time Performance Figure 4 shows the average computation time per step of the gradient-based method compared to that of the iterative open-loop approach. The time that is required to compute the optimal heading for the initial layout is not included. Note that even when using the brute-force approach to compute the optimal sequence, the gradient-based method is still about 50 times faster than the iterative open-loop approach even with 5 evaders. With the sequence pruning heuristics, the gradient-based method is more than 100 times faster than the iterative open-loop approach. One might notice that for 2 to 4 evaders, the heuristics with a branching factor of 2 takes about as long or even longer to compute than the brute force approach. This is due to some computational overheads that are necessary for the sequence heuristics to keep track of the already evaluated states. These computational overheads become less and less significant as the number of evaders increases. For 5-evader cases, the heuristics with $K = 2$ already outperforms the brute-force All-Seq approach in computational time. Similar to the survival time performance, there is not a significant difference between the computation time performance of the three different sequence pruning heuristics.

CONCLUSION & FUTURE WORK

We have in this paper proposed a gradient-based approach to solve the problem of evasion as a team against a fast pursuer, as an extension of our previous open-loop formulation of the pursuit-evasion game. By comparing with the iterative open-loop scheme introduced therein, we have demonstrated that the gradient-based approach achieves similar performance for the team of evaders at significantly less computational cost. Moreover, the gradient-based approach has the advantage of being able to leverage the benefits of distributed computations. Therefore, this gradient-based method, besides its novelty, proves to be a very promising computational approach to efficiently deal with the complexity in cases of a large number of evaders.

For future work, we would like to incorporate uncertainty of the pursuer's location into our formulation. This will make the model more realistic since in reality, it is hardly possible for the evaders to obtain the exact state information of their opponent. Moreover, the optimal strategy for the pursuer against the evaders employing the iterative open-loop scheme is still an open problem. Currently, the pursuer's strategy is an approximation of its best strategy. It would be interesting to compute the pursuer's optimal control against the iterative open-loop evaders, though we conjecture that the computation will be intractable as the space of possible pursuer's trajectories grows exponentially with number of open-loop iterations performed on the evaders' side. However, possibilities do exist that there is an intelligent way of cutting the search space significantly down to a polynomial size of the number of open-loop iterations made by the evaders, hence rendering the computation tractable.

REFERENCES

[1] Garvey, J., Kehoe, B., Basso, B., Godwin, M., Wood, J., Love, J., Liu, S. Y., Kim, Z., Jackson, S., Fallah, Y., et al., 2011. "An autonomous unmanned aerial vehicle system for sensing and tracking". In Infotech@ Aerospace Conference.

[2] Shevchenko, I., 1997. "Successive pursuit with a bounded detection domain". *Journal of Optimization Theory and Applications,* **95**(1), pp. 25–48.

[3] Abramyants, T. G., Ivanov, M. N., Maslov, E. P., and Yakhno, V. P., 2004. "A detection evasion problem". *Automation and Remote Control,* **65**(10), pp. 1523–1530.

[4] Abramyants, T., Maslov, E., and Yakhno, V., 2008. "Evasion of multiple target in three-dimensional space". *Automation and Remote Control,* **69**(5), pp. 737–747.

[5] Fuchs, Z., Khargonekar, P., and Evers, J., 2010. "Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game". In 2010 49th IEEE Conference on Decision and Control (CDC), pp. 3091 –3097.

[6] Shevchenko, I., 2004. "Minimizing the distance to one evader while chasing another". *Computers & Mathematics with Applications,* **47**(12), June, pp. 1827–1855.

[7] Breakwell, J. V., and Hagedorn, P., 1979. "Point capture of two evaders in succession". *Journal of Optimization Theory and Applications,* **27**(1), Jan., pp. 89–97.

[8] Chikrii, A. A., and Kalashnikova, S. F., 1987. "Pursuit of a

group of evaders by a single controlled object". *Cybernetics and Systems Analysis,* **23**(4), pp. 437–445.

[9] Chikrii, A. A., Sobolenko, L. A., and Kalashnikova, S. F., 1988. "A numerical method for the solution of the successive pursuit-and-evasion problem". *Cybernetics and Systems Analysis,* **24**(1), pp. 53–59.

[10] Belousov, A., Berdyshev, Y., Chentsov, A., and Chikrii, A., 2010. "Solving the dynamic traveling salesman game problem". *Cybernetics and Systems Analysis,* **46**(5), pp. 718–723.

[11] Berdyshev, Y., 2002. "A problem of the sequential approach to a group of moving points by a third-order non-linear control system". *Journal of Applied Mathematics and Mechanics,* **66**(5), pp. 709–718.

[12] Berdyshev, Y., 2008. "On a nonlinear problem of a sequential control with a parameter". *Journal of Computer and Systems Sciences International,* **47**(3), pp. 380–385.

[13] Berdyshev, Y., 2011. "Choosing the sequence of approach of a nonlinear object to a group of moving points". *Journal of Computer and Systems Sciences International,* **50**(1), pp. 30–37.

[14] Serov, V. P., 2000. "Optimal feedback strategy in the game variant of generalized travelling salesman problem". *Proceedings volume from the 11th IFAC Control Applications of Optimization Workshop,* **2**, pp. 635–640.

[15] Liu, S.-Y., Zhou, Z., Tomlin, C., and Hedrick, K., 2013. "Evasion as a team against a faster pursuer". *Proceedings of the 2013 American Control Conference (ACC2013)*, June.

[16] Polak, E., 1997. *Optimization: Algorithms and Consistent Approximations*. Springer-Verlag.

[17] Brayton, R., Director, S., Hachtel, G., and Vidigal, L., 1979. "A new algorithm for statistical circuit design based on quasi-newton methods and function splitting". *IEEE Transactions on Circuits and Systems,* **26**(9), Sept., pp. 784 – 794.