Thoughts on Multi-Agent Robotic Surveillance Problems

Connor Hughes

May 9^{th}, 2022

Background:

Let us consider a robotic surveillance problem with an omniscient intruder wherein our aim is to maximize the minimum capture probability, similar to that described in [1], but with a set of M robotic agents with identical capabilities sharing the surveillance task. The natural way to extend the omniscient intruder model from [1] is to assume that the intruder knows the locations of all agents at every time step, along with every agent's strategy. We say that "capture" occurs if any of the agents arrive at the intruder's location while an attack is underway. Thus, the capture probability is the likelihood that one or more agents arrive at the intruder's location within τ time steps.

Naively, we could assume that each of the agents has a strategy which is independent of the strategies and locations of the other agents. However, a more realistic formulation would involve the sharing of information between agents and/or with a central controller. Naturally, before we can develop an approach for strategy optimization, we must specify which agents can communicate with each other and what information they can share. The two key pieces of information which each agent can provide are their current location and their strategy. If we assume that the stochastic strategy for each agent is not time-varying, then the strategies can be shared once initially and only the updated agent locations must be shared at every time step.

In the case where each agent can communicate with every other agent or with a central controller, we can think about surveillance strategies from two different perspectives. One is to consider each agent having their own strategy, which may involve conditioning on the locations and strategies of other agents. The other perspective involves specifying one strategy for the entire team of agents, which implicitly specifies the strategy for each individual agent by the Law of Total Probability. As we will argue in the following sections, if all agents have identical capabilities, taking the viewpoint that a single team-wide strategy is specified by a central controller provides a general and useful framework (see "Comments on heuristic strategies" and "Comments on generality"). It is also more intuitive, as it still allows us to formalize the problem as a Stackelberg game with only two players: an intruder and a centralized agent-controller. Thus, in modifying the Stackelberg game from the single-agent setting, we must change the action space of the entity defending the graph but do not need to alter the number of players, the information structure, or the noncooperative nature of the game. Lastly, the use of a central controller and team-wide strategy may demand significant communication bandwidth and introduce a single point-of-failure, but there are a variety of practical applications for which these concerns are not critical (see "Comments on practical applications"). In the next section, we describe a framework for modeling team-wide strategies with Markov chains.

Markov chain-based team-wide strategy formulation:

Let us define the "state" of the game to be the set of current locations of the M robotic agents, and denote the corresponding state space by S. At time step k, the locations of the agents are represented by some state $s_k \in S$. We can define a set of "capture states" S_{cap} which include all of the possible sets of agent locations wherein one or more agents occupy the same location as the intruder during an attack. For each state s_k, we can specify transition probabilities associated with every possible collection of agent locations at time step $k+1$ (i.e., every feasible next state). Thus, strategies can still be modeled as Markov chains just as in the single agent case, but these Markov chains now exist within a new higher-dimensional state space. The capture probability associated with a given state s_k and a given strategy is the likelihood that the first hitting time from s_k to any state in S_{cap} is less than or equal to τ time steps. While this formulation dramatically increases the dimension of the problem, as S contains n^M states, it has the advantage of generality, in that every team-wide strategy which can be generated by specifying strategies for individual agents and allowing communication between them can be equivalently expressed with this framework. Thus, we are assured that the globally optimal surveillance strategy is within the set of strategies which can be specified. See the following sections for further discussion.

Comments on heuristic strategies:

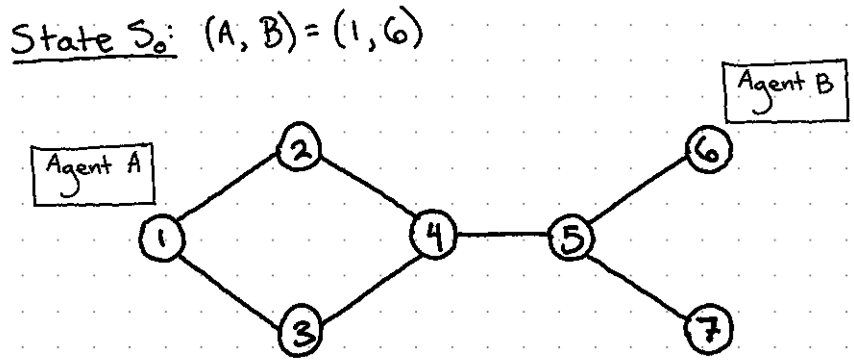
It is intuitive to consider certain heuristic approaches for strategy optimization in multi-agent robotic surveillance problems. For example, partitioning the graph into subgraphs and assigning the surveillance of each subgraph to an individual agent seems likely to be effective for some environment graphs, such as those which resemble a barbell graph. This approach splits the problem into two subproblems: determining the optimal way to partition the graph, and determining the optimal surveillance strategy for each agent on each subgraph. However, it is not evident that a partitioning-based approach will allow us to express the globally optimal strategy among the set of all strategies which the team of agents could implement. For other graph topologies, such as those which closely resemble a cycle graph, it may seem desirable to have every agent cyclically traverse the entire graph, and to vary the spacing between the agents in an unpredictable manner as they do so. Once again, though, there is no guarantee that such an approach will allow us to express the globally optimal team-wide strategy.

The issue is that selecting a strategy by a heuristic approach inherently restricts the space of team-wide strategies which can be considered. This restriction is also a key benefit, as long as we do not unknowingly use an approach which restricts us to considering only relatively ineffective strategies. It may be difficult to determine whether this is the case for an arbitrary environment graph. As such, there is value in a framework which allows for the expression of *all* strategies which the team of agents could implement, like that described in the previous section. Even if we are unlikely to find the globally optimal strategy with this framework, we at least know it is theoretically possible. Thus, this framework enables us to use an optimization algorithm to identify team-wide strategies that could serve as valuable performance benchmarks when evaluating heuristic-based strategies. In addition, optimizing over the entire space of team-wide strategies may yield effective and unexpected strategies that reveal insights and lead to improved heuristics.

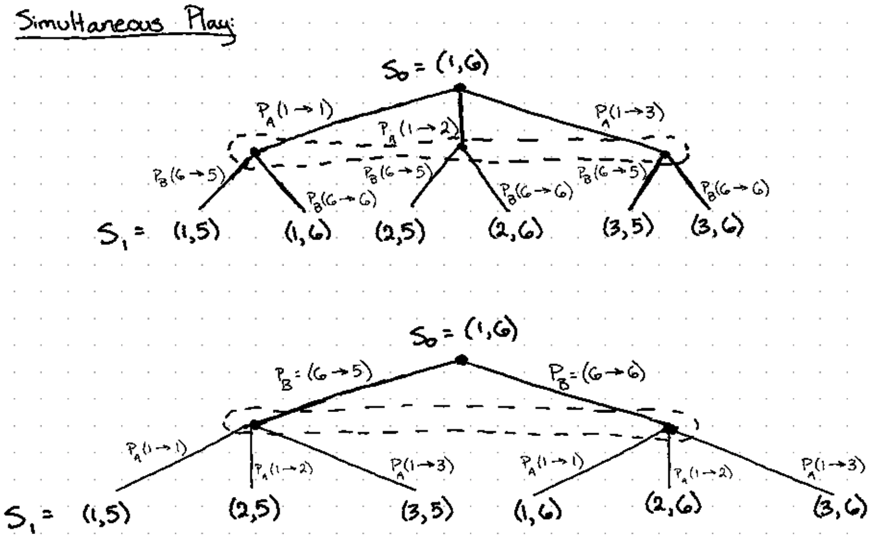
Comments on generality:

If we take the perspective that the agents each have their own strategy in the setting where each agent can communicate with every other, in the most general formulation, each agent should be able to condition their strategy on the locations and strategies of all other agents. However, there is an additional question of whether the agents should realize their next move simultaneously or sequentially at each time step. That is, from a game-theoretic perspective, should their strategies be based on a framework with "simultaneous plays" or "alternating plays".

For example, suppose we have only two agents, agent A and agent B, navigating a shared graph, sketched below. At a given time step, agent A is at node 1 with the opportunity to move to nodes 1, 2, or 3 at the next time step, and agent B is at node 6 with the opportunity to move to nodes 5 or 6 at the next time step. We denote the initial state of the pair of agents (which for clarity we will refer to in this section as the initial "team state") by s_0, and denote the team state after 1 time step by s_1.

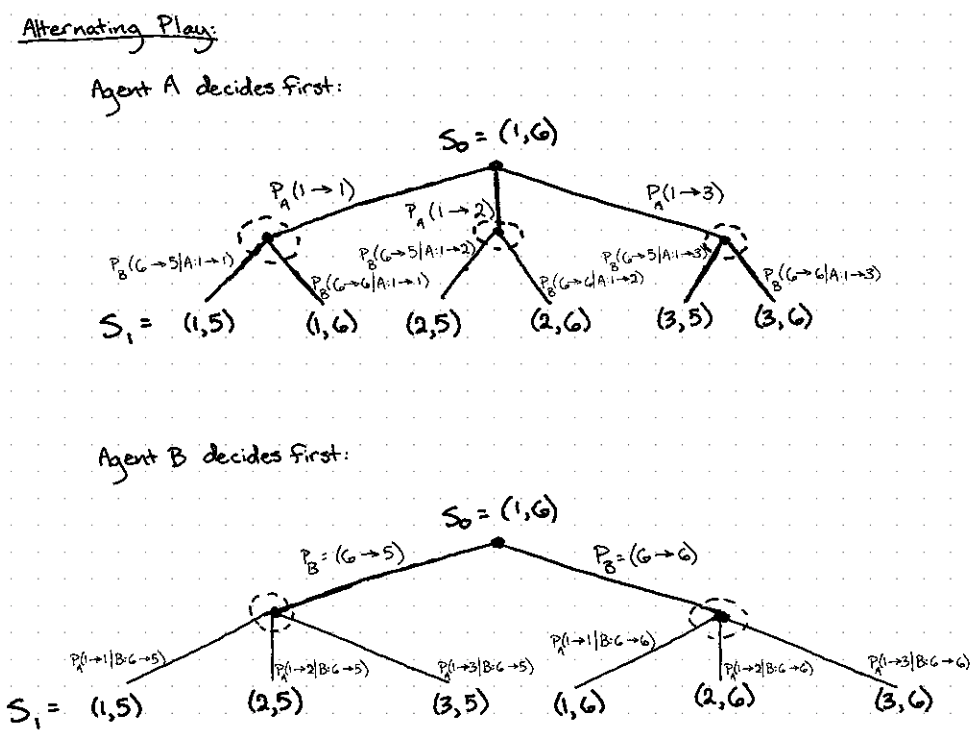


In the simultaneous play setting, agents A and B each have a strategy which gives some probability of transitioning to each of the possible nodes at the next time step. For both agents, these probabilities can be conditioned upon the other agent's location and strategy. However, each agent must realize a state transition based on their strategy without knowledge of the state transition realized by the other agent at this time step. Let's assume we can identify an optimal set of transition probabilities for each agent in this scenario by an iterative method, such as iterated best response. We will denote the probability that agent A transitions from node i to node j by $P_A(i \to j)$, and similarly for agent B. We can then represent one time step as a single stage in a game with simultaneous play, as shown in two equivalent ways in extensive form below.



Notice the information set shown by the dotted line represents one agent's knowledge about the other agent's next state. Notice also that there are 6 possible team states which the pair of agents can enter at the next time step (after both agents have moved), and 5 total transition probabilities for the two agents. Given that each agent's transition probabilities must sum to 1, we see there are actually only 3 parameters here which we can freely specify. That is, the set of 6 "team state transition probabilities" $P(s_1 = (i,j)|s_0 = (1,6))$ lives in a subspace of dimension 3 within the probability 6-simplex.

In the alternate play setting, either agent A or agent B "plays first". This is analogous to one agent realizing a state transition based upon their strategy, then announcing their next move to the other agent. This enables the agent who plays second to condition their transition probabilities on the first agent's *next* state. These two scenarios are shown in extensive form below.



Notice the difference in the information sets between this version of the game and the simultaneous play version. In the case where agent A plays first, they can specify 3 transition probabilities (with 2 degrees of freedom) and agent B is able to specify 6 transition probabilities (with 3 degrees of freedom). In the case where agent B plays first, they can specify 2 transition probabilities (with 1 degree of freedom), and agent A can specify 6 transition probabilities (with 4 degrees of freedom). In either case, the set of 6 team state transition probabilities $P(s_1 = (i,j)|s_0 = (1,6))$ depends on 5 free parameters between the two agents. Thus, the space of team state transition probabilities which can be prescribed is the entire 6-simplex. So, we see that any set of team state transition probabilities which can be realized under the simultaneous play formulation can also be realized under the alternating play formulation.

Under the assumption that all agents have identical capabilities, we do not care which of the agents arrives at the intruder's location. So, at each time step, we do not care *which* agent is at a given location, only *whether* there is an agent at a given location. Thus, the "team state" contains all the essential information with respect to capture, and the team state transition probabilities fully define the capture probability. So, we can see that it does not matter which of the agents plays first in the alternating play formulation. As long as we assume that the agent which moves first *knows* that the second agent will condition their transition probabilities on the first mover's next state, we expect the first-moving agent will set their own transition probabilities so as to enable the team to achieve the optimal set of team state transition probabilities. That is to say, we do not care which agent plays first, because we assume that since their objectives are perfectly aligned they will be able to collaborate, and determine how to generate the optimal team state transition probabilities.

So, while it would be possible to describe every team-wide strategy by establishing an order-of-play for the agents and prescribing conditional strategies for each agent as above, this is not an efficient way to represent the team strategy. A more efficient approach would be to establish the order-of-play based upon the current locations of the agents, which allows for storing simpler strategies for each agent. However, updating the order-of-play at each time step and realizing state transitions accordingly would be complicated to implement, and is not necessary to achieve the desired team state transition probabilities. In this example, the team-wide strategy formulation described earlier allows us to directly specify the probabilities $P(s_1 = (i,j)|s_0 = (1,6))$. In doing so, we also implicitly specify strategies for each agent by the Law of Total Probability. For example, we see that $P_A(1 \to 3) = P(s_1 = (3,5)|s_0 = (1,6)) + P(s_1 = (3,6)|s_0 = (1,6))$. Also, specifying a team-wide strategy in this scenario enables us to directly set 6 team state transition probabilities, with 5 degrees of freedom, comprising the entire 6-simplex just as in the alternating play formulation.

We conclude with a few final comments on the generality of the team-wide strategy formulation. In this formulation, we have made the assumption that every agent is able to communicate with a central controller. This assumption encompasses the broadest possible set of team-wide strategies, as it allows for maximal sharing of information and assumes no limitation on available computational power. With this assumption, we are still able to replicate strategies involving communication between subsets of the agents, consensus algorithms, limited computation power, etc, but we can also specify strategies which are not feasible under these limitations. On another note, we might suppose that it is optimal in certain scenarios to have one or more agents use deterministic strategies while the others use stochastic strategies. Fortunately, the set of deterministic strategies for any collection of agents is a subset of the space of stochastic strategies for those agents, so any partially- or fully-deterministic team-wide strategy can also be specified with the team-wide strategy formulation. Lastly, note that any heuristic-based strategy which can be implemented by specifying individual agents' strategies can also be specified in a team-wide manner. For example, partitioning-based approaches might be implemented by giving individual agents independent strategies with zero transition probabilities for traversing any edge which leaves their assigned subgraph. Every possible departure of each agent from their respective subgraph also corresponds to a set of team state transitions, which can also have probabilities set to zero to prevent this from occurring. Furthermore, leveraging the Law of Total Probability to implicitly specify individual agent strategies in a team-wide manner does not prohibit those strategies from being independent from one another.

Comments on practical applications:

With regard to practical applications, note that as the number of agents becomes very large, specifying a surveillance strategy becomes a swarm-control problem, wherein our options may be limited by constraints on communication bandwidth and the desire to avoid introducing a single point-of-failure via a centralized control system. In this setting, distributed algorithms which leverage only local information from nearby agents are highly desirable. However, it seems there are numerous real-world applications wherein it would be useful to coordinate a small enough team of agents that bandwidth constraints are not critical, and wherein we are not concerned with the adversary's ability to target and destroy a centralized controller. In particular, if we specify a team-wide strategy which is not time-varying, then this strategy can be optimized offline and stored by each agent. This offline optimization is vital, as the optimization process is likely to be very computationally expensive due to the size of the state space associated with the agents' locations. However, supposing each agent can store the optimized strategy, the only information which must be shared at each time step is the current set of agent locations.

There are examples of Stackelberg security games being used to develop strategies which have already been implemented in the real world, such as the planning of drug inspections by the TSA at Los Angeles International Airport, and the deployment of Federal Air Marshalls to domestic flights in the US [2]. Furthermore, there already exists technology which allows for the real-time tracking of the locations of agents/resources in military and policing applications, which could be used to implement a predetermined team-wide surveillance strategy. In other scenarios, such as planning coast guard patrols for drug trafficking intervention in the Channel Islands region, it seems likely that the number of agents patrolling at any given time might be sufficiently small that optimizing team-wide strategies using the framework suggested here is computationally tractable. Additionally, in some real-world applications, the number of agents and the size of the graph are matters of scale which can be selected via partitioning to make computation feasible. For example, consider the task of deploying police patrols throughout the city of San Francisco. There are approximately 2,000 police officers in the SF Police Department, though a study of 61 police departments suggests that on average only 2/3 of these officers are assigned to patrolling tasks [3]. Also, we should not expect that they will all patrol the city at the same time. Suppose at any given moment there are about 400 police officers on patrol throughout San Francisco. At the broadest level, we might hope to be able to identify a strategy without partitioning, which specifies a strategy for all 400 patrolling police officers without restricting where they might patrol within the city. However, if this is infeasible, we can simplify the problem by partitioning at varying levels of resolution. We do not have to partition the entire city into 400 small regions such that a single police officer is assigned to every region, but rather, we can partition the city somewhat more broadly and assign a small team of officers to patrol each region. The size of the regions and number of assigned officers can be adjusted to ensure feasible computation, then optimized strategies for each region can be compared with an analysis of existing patrolling strategies used by the police force.

References

[1] X. Duan, D. Paccagnan, and F. Bullo, "Stochastic strategies for robotic surveillance as stackelberg games," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 769–780, 2021.

[2] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe, "Stackelberg security games: Looking beyond a decade of success," *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 5494–5501, 2018.

[3] J. McCabe, "An analysis of police department staffing: How many officers do you really need?," tech. rep., ICMA Center for Public Safety Management.