Multi-Sweep: Seeker Strategies to Capture Hiders with Moving Obstacles

Ayushman Panda[1] and Kamalakar Karlapalem[1]

[1]International Institute of Information Technology, Hyderabad
ayushman.panda@research.iiit.ac.in
kamal@iiit.ac.in

Abstract. The classic hide-and-seek game serves as an abstraction for many real-world scenarios like capturing intruders in a closed space, locating objects, patrolling an area, etc. Since most of the present work is based on static obstacles, we address solutions for the hide-and-seek game in an environment where the obstacles are not static. We evaluate and improve the existing strategies provided for capturing hiders in an environment with moving obstacles. We design strategies that would facilitate seekers to capture hiders in a similar environment. We simulate two strategies and compare their performance in different scenarios. The 4-way sweep strategy captures all hiders faster than the 2-way sweep strategy in any given environment but demands more seekers.

Keywords: Strategies · Hide and seek · Game · Surveillance · Coverage Point · Strategic Point · Moving Obstacles

1 Introduction

We model an environment where we introduce agents and obstacles to simulate the hide-and-seek game. Tandon and Karlapalem [14] studied the hide-and-seek game in a 2D environment with static obstacles. They introduced abstractions like coverage and strategic points to encapsulate environmental properties for their proposed strategy. Our work uses those abstractions and introduces some more to extend the work to a dynamic environment where obstacles are moving. The agents are divided into hiders and seekers. The hiders use the hiding spots around obstacles to stay hidden in the environment while the seekers search for hiders and eliminate them. The seekers follow a strategy to search for hiders and eliminate them from the environment. The obstacles move independently in the environment, which makes it challenging for the seekers to eliminate hiders hiding around them. The environment, agents and obstacles have individual set of properties which is followed throughout the simulation of the game. All agents and obstacles follow the defined properties to interact with the environment and with each other. The seekers have no information on the position of hiders in the environment. The properties of the environment, agents, and obstacles are given in section 2 of the paper.

The paper contributes two variations of sweep strategies for the seekers to capture hiders, namely *2-way sweep strategy* and *4-way sweep strategy.* Both the strategies are simulated and the performance of strategy is compared based on number of steps taken to eliminate all hiders in the given environment.

2 The Environment

We are given an environment **E.** The environment consists of obstacles and agents, both of which can move in the environment. **E** is divided into a $M \ X \ N$ grid. Each cell in the grid is a pixel in the screen for the simulator. Every obstacle and agent in the environment is centered in a specific pixel (i, j) in the grid, $0 \leq i < M$ & $0 \leq j < N$. The size of the obstacle and agents would define the number of pixels taken by it on the screen.

2.1 Obstacles

Obstacles (O) are abstraction for objects/blocks which are arbitrarily placed in the environment. The obstacles have a square geometry and are centered at a given cell in the grid. The obstacles move independently in the environment to occupy neighbouring cells. The agents cannot move the obstacles.

Obstacles block the field of vision of agents. Moving obstacles can help the hiders in increasing their hiding time in the environment, while it is a trouble for the seekers since the moving of obstacles increases the elimination time of hiders by sheltering them. Obstacles are 2-dimensional blocks in the environment E. Since an obstacle blocks the field of vision of agents, a seeker can never keep track of all edges of the obstacle at once. Hence, hiders associated with the exposed surface of an obstacle can only be caught and eliminated by a seeker. Seekers may however form strategy to surround an obstacle from all sides and put all corresponding edges into their combined field of vision, ultimately capturing all hiders associated with that obstacle. This would require a large number of seeker agents for each obstacle. The multi-sweep strategies defined in section 4 present a better way to trap obstacles from different sides with less number of seekers.

Strategic Points Strategic Points (SP) are spots in the environment where hiders can hide. Strategic points are defined on the mid point of each edge of an obstacle. Since the obstacles have a square geometry, Four strategic points are associated with any obstacle.

2.2 Coverage Points

Coverage points (CP) are seeking points for seekers in the grid, where seekers can take position to capture hiders. The seekers follow the sweep strategy which computes the coverage points for the seekers based on the current state of the environment. The number and position of coverage points changes throughout the game. Sweep strategy demands one seeker to be placed at each coverage point. We evaluate 2-way sweep strategy and 4-way sweep strategy based on the number of seekers available in the game and the median number of game completion steps required.

2.3 Agents

Agents are mobile autonomous individuals which belong to the environment. They are classified as hiders and seekers. Hiders are intruders in the environment and want to stay hidden in the environment as long as possible. Seekers are guardians of the environment who want to eliminate all hiders from the environment in a minimum number of steps.

The hiders and seekers take random position in the environment initially. To simulate hiders goal of staying hidden longer in the environment, we update hiders position according to the following probability distribution:

- Remain in the same strategic point with a probability of P_1
- Traverse to a nearby strategic point with a probability P_2
- Traverse to a nearby cell in the environment which is not a strategic point with a probability of P_3

The *field of vision,* $\mathbf{v}$ of an agent is the number of cells an agent can track from the current position. If an obstacle comes in the visibility range of any agent, the agent can track only the strategic points which faces towards the agent's direction. The field of vision of an agent is hence blocked by obstacles. A cell which falls under the field of vision of an agent is called *visible adjacent cell.*

3 Hide-and-Seek Game

The game starts with dividing the Environment E into an MXN grid. Obstacles (O) are arbitrarily placed in the grid. Hiders are allowed to take random position in the environment first. The seeker team chooses a strategy and decides the required number of seekers (NS). The seekers then occupy the coverage points given by the strategy. The game continues until all hiders are caught or maximum allowed game steps is reached. [see section 3.1].

3.1 Game Completion

We use a limit $\boldsymbol{L}$*,* which denotes the maximum number of game steps allowed for both agents. The verdict of the game (all hiders are caught or not) is checked at the end of L game steps.

- *Seekers victory*: If the seekers eliminate all hiders from the environment before L game steps, the seekers declare victory, and the game terminates.
- *Hiders Victory* : If all hiders are not caught at the end of L game steps, the seekers lose. The hiders declare victory, and the game terminates.

4 Multi-Sweep strategies for Moving Obstacles

Sweep strategy [8] works on the principle of *Trap Strategy [15]*, but on individual obstacles instead of *hiker graph* [15]. Sweep strategy ensures trapping hiders in the environment by collapsing the available environment for hiders through a surveillance technique, which limits the available hiding places for the hiders.

We extend the technique used in sweep strategy to design two more strategies: *2-way sweep strategy* and *4-way sweep strategy*. These *Multi-Sweep* strategies guide seekers to traverse through the environment from different corners of the environment simultaneously instead of just one.

Taking the example of 2-way sweep strategy in a square environment with side s and visibility set to 1:

- The seekers start searching for hiders from two opposite corners of the environment.
- The seekers capture all the hiders associated with any obstacle which falls under their visibility range. Since v=1 in this example, the seekers capture hiders present in their own cells. Similarly, if v=x, the seekers can capture hiders from x adjacent grid cells.
- When an obstacle is encountered by the seeker team, the seekers look for hiders in the strategic points of the obstacles which fall under their field of vision. All hiders associated with those strategic points are eliminated from the environment.
- In the next step, seekers occupy the next series of un-visited grid cells. Seekers go around the obstacles while sweeping.
- Seekers keep occupying new cells until seekers from each corner reach the diagonal. When all seekers reach the diagonal, the seeker teams repeat the sweep strategy again until all hiders are caught.

In the 4-way sweep strategy, we deploy seekers to traverse from all four corners of the rectangular environment.

Summarising Multi-Sweep Strategy Consider a square environment with side a and visibility(v) of agents set to 1.

- *Number of Coverage Points required for Sweep Strategy*: The maximum number of Coverage points required during the game is the number of seekers required to keep track of the longest sequence of cells for sweep in the grid (diagonal). The number of coverage points required in a 4-way sweep strategy remains constant, i.e., **2a**. This is because we optimized the traversal plan for the seekers by guiding two seeker teams to traverse toward the diagonal while the other two seeker teams traverse away from the diagonal.
- *Role of Offset(v)*: Since offset defines the visibility range of seekers, an increase in offset value would require less number of seekers to inspect cells in the grid.
- *Game completion time*: The game completion time is dependent on the number of steps required to cover the entire grid. In our example, a square environment with each side a has its diagonal as the longest sequence of cells in the

grid. Since $v = 1$, agent's visibility is limited to the cell they are present on. So, we need as many seekers as there are cells in the diagonal. The seeker team takes a steps to reach from the corner to the diagonal of the environment.
- It is to be noted that the hiders might remain hidden in a strategic point that does not fall under the seeker's field of vision during the course of traversal. The obstacles block the seeker's vision, and the hiders associated with strategic points which do not fall under the seeker's visibility region remain hidden in the environment. The seekers hence repeat their traversal route multiple times until all hiders are caught.

5 Simulation

We simulate the *multi-sweep strategies* with two fairly large 2-D environment maps (MAP-1 and MAP-2). The properties of each environment is listed in table 1. The maps provide a bounded playing area for the hide-and-seek game. The simulation includes all the spatial features and abstraction from the defined environment [see section 2]. The simulation occurs in terms of discrete game steps. In each step of the game, every agent and obstacle is expected to take action. Since the movement of agents is based on a strategy and obstacles may choose to remain associated in the same cell, staying in the same cell is a valid action for both agents and obstacles. The probability distribution for hider movement was set to 0.1, 0.7, and 0.2 for P_1, P_2 and P_3 respectively [see section 2.3].

Environment Properties				
	Dimension	Cells	Obstacles	Hiders
MAP-1	100x100	10000	500	300
MAP-2	500x500	250000	2000	2500

Table 1. Properties of MAP-1 and MAP-2

Both the maps differ on the basis of grid size and number of obstacles present. Each obstacle has four strategic points associated with it. We run the game for each map-strategy pair with a significantly high value of **L**. The high value of L resulted in the seeker's victory in every round. Since the main idea of the simulation is to evaluate the seeker's strategies based on the game completion steps, we choose a high value for L. We run 50 game-rounds each for 2-way sweep strategy and 4-way sweep strategy.

5.1 Results

The 2-way sweep strategy demands a maximum of **200** seeker agents for MAP-1, while **1000** seekers for MAP-2, to capture all hiders. We analyze the result of

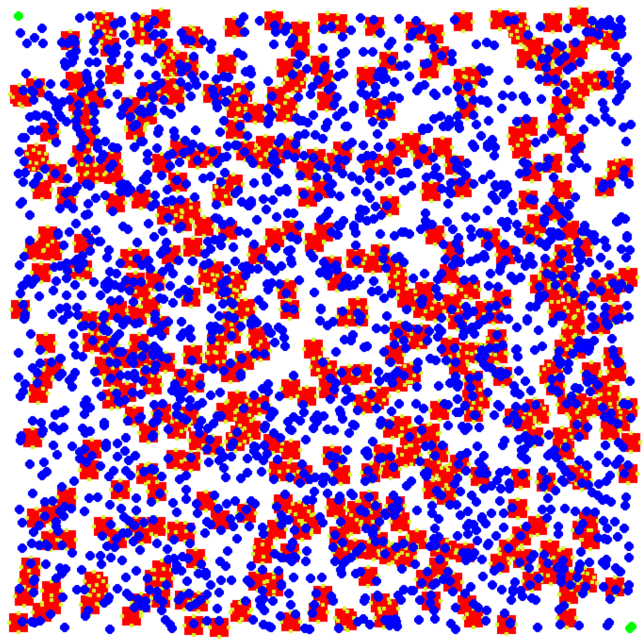

Fig. 1. Map with 500 obstacles and 2000 hiders in a 650 x 650 environment. Obstacles are shown in red, hiders in blue, strategic points in yellow and seekers in green.

the strategy by running 50 rounds of the game. 4-way sweep strategy works with a constant demand of 200 and 500 seekers, respectively, for MAP-1 and MAP-2.

From the median game completion time, we conclude that the multi-sweep strategies guarantee hider's elimination in a significantly fewer number of steps as compared to the set-cover strategy [8] and sweep strategy [8]. In the 2-way sweep strategy, we provide the maximum number of seekers as required by the environment based on the grid size. However, we need the maximum number of seekers only when both the seeker teams reach the diagonal from their respective corners. Once they traverse toward the corner, the demand for seekers keeps decreasing. This produces idle seeker agents who wait for the seeker team to move away from the center so that they can join them again. The 4-way sweep strategy performs better logistically since there would be no idle seekers as in the case of the 2-way sweep strategy. Opposite traversal of seekers ensures that the idle agents of one set of seekers who are traversing away from the diagonal are engaged in another seeker team that is traveling towards the diagonal. This ensures a constant number of seeker demand and removes the possibility of any idle agents.

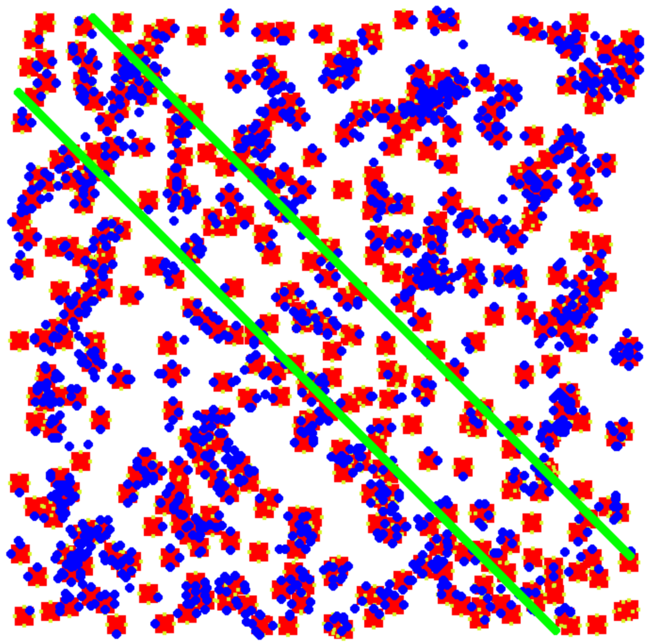

Fig. 2. Seekers following 2-way sweep strategy

Game Completion Time The game completion steps for the sweep strategy across 50 game rounds with random obstacle movement in maps 1 and 2 are given in Table 2 and Table 3. The sweep strategy is observed to perform well even in large environments considering the lower number of seeker requirements as compared to the set-cover strategy [8].

5.2 Comparison of Strategies

We compare both strategies in map 1 and 2 with different obstacle movement techniques. The game completion time for each map is listed in Table 2 and Table 3.

Random Obstacle movement: While the set-cover strategy takes less game steps to eliminate hiders from the environment even with partial seekers available, it is to be noted that the number of seekers demanded by the set-cover strategy is larger than sweep-strategy. So, sweep strategy is more efficient in case of random obstacle movement in any given environment.

Hider-friendly Obstacle movement: The seekers take more game steps to eliminate all hiders in case of hider-friendly obstacle movement, as expected. However, the sweep strategy takes significantly fewer steps than the set-cover strategy in both the environment to capture and eliminate all hiders. This is

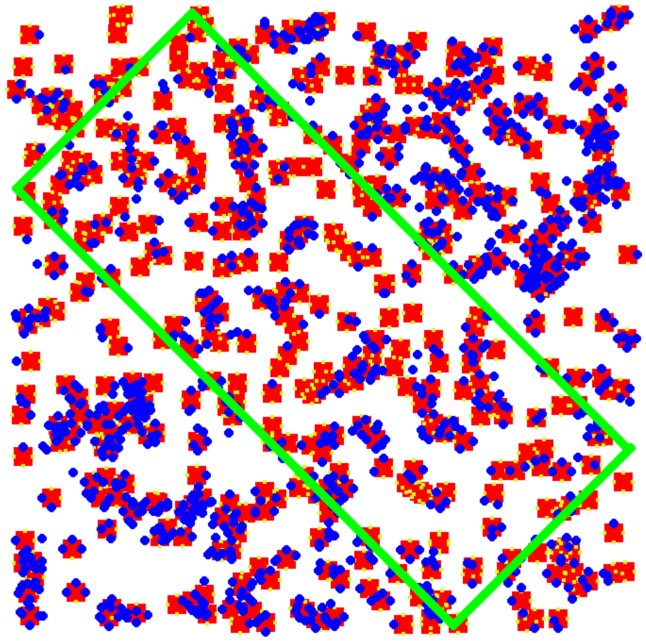

Fig. 3. Seekers following 4-way sweep strategy

because the obstacles cannot escape to hider-friendly cells since the traversal of seekers in the sweep strategy ensures that no gap nodes are present.

Seeker-friendly Obstacle movement: The seekers capture hiders in significantly fewer game steps in any given round of the game. While the set-cover strategy works well in a smaller environment like MAP-1, the sweep strategy outperforms in the case of larger environment, as in MAP-2. Considering the lower number of seeker agents demand with almost similar game completion steps as the set-cover strategy, the sweep strategy would be an ideal strategy to follow in case of large environments.

The set-cover strategy takes almost four times of game completion steps in the case of hider-friendly obstacle movement as compared to seeker-friendly obstacle movement in both maps. The sweep strategy, however, takes twice the number of game completion steps in the case of hider-friendly obstacle movement in comparison to seeker-friendly obstacle movement. The sweep strategy can be followed in large environments. It maintains the overall balance of less number of seekers and lower game completion steps to be considered an ideal strategy in most environments.

The multi-sweep strategies leverages the performance of the sweep strategy and improve it many folds. The simulation is helpful in visualizing the course of the game from each agent's point of view. From Figure 4 We observe that the type of obstacle movement has no impact on the median game completion steps

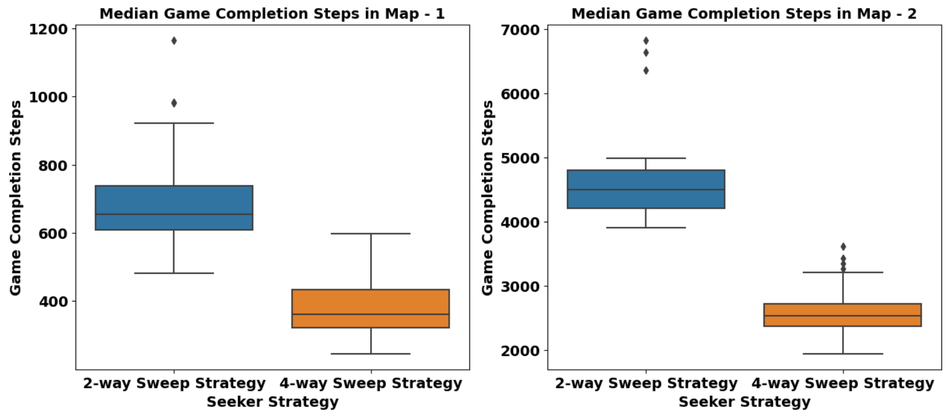

Fig. 4. Median Game Completion Steps taken by Seekers following Multi-Sweep Strategies in MAP-1 and MAP-2 with random obstacle movement.

in our simulation. The reason for this behavior is no place for obstacles for hider-friendly movements in the environment, and hence any direction an obstacle move is going to be traversed by the seeker team. The hiders are hence trapped in the obstacle when seekers invade from more than one direction simultaneously. We also observe that the median game completion steps for the 4-way sweep strategy are close to half of what the 2-way strategy takes to eliminate all hiders from the environment. Since we have set a very high value of $\boldsymbol{L}$, we let the seekers take as many steps as required to catch all hiders. The outliers in the median game completion times shown in Figure 4 can be considered as cases where the seeker team failed to catch all hiders in an environment with a strict constraint on game completion steps.

The code for the simulator can be found here: https://github.com/AYUSHMAN-PANDA/NanSim and a video demonstration of the working of the simulator can be found here: https://youtu.be/5vXFnogcGoU

Obs. Mov.	**2-way Sweep(200)**	**4-way Sweep(200)**
Random	656	362
Hider Friendly	671	379
Seeker Friendly	648	368

Table 2. Median Game Completion Steps in MAP-1 across 50 rounds of the hide-and-seek game. 2-way sweep required a maximum of 200 seekers, while 4-way sweep constantly demanded 200 seekers.

Obs. Mov.	**2-way Sweep(1000)**	**4-way Sweep(1000)**
Random	4506	2542
Hider Friendly	4711	2678
Seeker Friendly	4512	2546

Table 3. Median Game Completion Steps in MAP-2 across 50 rounds of the hide-and-seek game. 2-way sweep required a maximum of 1000 seekers, while 4-way sweep constantly demanded 1000 seekers.

6 Related Work

The hide-and-seek game has been an interesting and well-known field of study for a long time. Besides being a game, hide-and-seek problem has found application in other domains like search theory in operation research, pursuit-evasion games with multiple mobile agents (Chung et al. [4], Parsons [9], Megiddo and Hakimi [7]), multi-agent security (Pita et al. [12], Fang et al. [5]), Paruchuri et al. [10], Paruchuri et al. [11]), multi-agent deep reinforcement learning (Baker et al. [3]), etc. The princess and monster game by Isaacs [6] introduced the hide-and-seek game with moving hiders. The problem was further studied by Alpern [1] with changing parameters and was finally solved for a generalized case by Alpern, and Gal [2]. Most recent work on hide-and-seek focuses on emergent agent behavior with the help of reinforcement learning and AI techniques. Strickland [13] proposed reinforcement learning-based strategies in a 3D environment where agents learned to capture hiders in challenging environments with moving obstacles and mobile agents. This is in contrast to our strategies which work on a 2D environment, based on a heuristic approach, and works on predictive traversal path planning. Tandon and Karlapalem [14] studied the hide-and-seek game in a 2D environment with static obstacles. They introduced abstractions like coverage and strategic points to encapsulate environmental properties for their proposed strategy. Panda and Karlapalem [8] explored the field of moving obstacles with set-cover and sweep strategies. Our work is a step further in improving the performance of sweep strategy in an environment with moving obstacles.

7 Conclusion

The two multi-sweep strategies presented in the paper perform well and can be followed in real-life simulations of agents in a given environment. The simulation and comparison of strategies in section 5 of the paper gives insight into the performance of individual strategies in various environments and scenarios. The strategies capture hiders in a reasonable number of steps with different types of obstacle movement. The strategies follow classical set-cover and traversal-based algorithms, which is less compute intensive and ideal for real-time strategic games, as compared to most of the present work, which provides reinforcement learning-based strategies, which demands extensive training data for the algorithm to *learn* strategies.

This paper is a significant improvement over the set-cover, and sweep strategy [8]. We evaluated the performance of the strategies with a larger number of obstacles and hiders. The hiders are allowed to take any position in the environment and not just strategic points. This takes care of stagnant conditions where the seekers would keep searching for hiders in the strategic points while the hiders are hidden in some other non-strategic point in the environment. The simulation also gives vital statistics while the game is in progress, which is helpful in keeping track of the progress of the game and the state of each agent in the game. Future work on this problem is on conditions when fewer than required seekers are available and strategies when seekers fail to eliminate hiders.

References

1. Alpern, S., Baston, V., Gal, S.: Network search games with immobile hider, without a designated searcher starting point. International Journal of Game Theory **37**(2), 281–302 (Jun 2008). https://doi.org/10.1007/s00182-008-0116-7, `https://doi.org/10.1007/s00182-008-0116-7`
2. Alpern, S., Gal, S.: The theory of search games and rendezvous. Springer (2006), `https://www.springer.com/fr/book/9780792374688`
3. Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., Mordatch, I.: Emergent tool use from multi-agent autocurricula (2019). https://doi.org/10.48550/ARXIV.1909.07528, `https://arxiv.org/abs/1909.07528`
4. Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics. Autonomous Robots **31**(4), 299 (Jul 2011). https://doi.org/10.1007/s10514-011-9241-4, `https://doi.org/10.1007/s10514-011-9241-4`
5. Fang, F., Stone, P., Tambe, M.: When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In: Proceedings of the 24th International Conference on Artificial Intelligence. p. 2589–2595. IJCAI'15, AAAI Press (2015)
6. Isaacs, R.: Differential games; a mathematical theory with applications to warfare and pursuit, control and optimization [by] Rufus Isaacs. Wiley New York (1965)
7. Megiddo, N., Hakimi, S.: Pursuing Mobile Hiders in a Graph. Discussion Papers 360, Northwestern University, Center for Mathematical Studies in Economics and Management Science (Dec 1978), `https://ideas.repec.org/p/nwu/cmsems/360.html`
8. Panda, A., Karlapalem, K.: Capturing hiders with moving obstacles. In: Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems. AAMAS '23, International Foundation for Autonomous Agents and Multiagent Systems (forthcoming)
9. Parsons, T.D.: Pursuit-evasion in a graph. In: Alavi, Y., Lick, D.R. (eds.) Theory and Applications of Graphs. pp. 426–441. Springer Berlin Heidelberg, Berlin, Heidelberg (1978)
10. Paruchuri, P., Pearce, J., Marecki, J., Tambe, M., Ordóñez, F., Kraus, S.: Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. pp. 895–902 (01 2008). https://doi.org/10.1145/1402298.1402348

11. Paruchuri, P., Pearce, J.P., Tambe, M., Ordonez, F., Kraus, S.: An efficient heuristic approach for security against multiple adversaries. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. AAMAS '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1329125.1329344, https://doi.org/10.1145/1329125.1329344
12. Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., Kraus, S.: Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. pp. 125–132 (01 2008). https://doi.org/10.1145/1402795.1402819
13. Strickland, E.: Ai agents play hide-and-seek: An openai project demonstrated "emergent behavior" by ai players - [news]. IEEE Spectrum **56**, 6–7 (11 2019). https://doi.org/10.1109/MSPEC.2019.8889898
14. Tandon, A., Karlapalem, K.: Medusa: Towards simulating a multi-agent hide-and-seek game. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 5871–5873. International Joint Conferences on Artificial Intelligence Organization (7 2018). https://doi.org/10.24963/ijcai.2018/866, https://doi.org/10.24963/ijcai.2018/866
15. Tandon, A., Karlapalem, K.: Capturing oracle guided hiders. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. p. 1350–1358. AAMAS '20, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2020)