Agent Strategies for the Hide-and-Seek Game

by

Akshat Tandon, Kamalakar Karlapalem

in

International Conference on Autonomous Agents and Multiagent Systems
(*AAMAS 2018*)

Stockholm, Sweden

Report No: IIIT/TR/2018/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2018

Agent Strategies for the Hide-and-Seek Game

Extended Abstract

Akshat Tandon
Agents and Applied Robotics Group
KCIS, IIIT-Hyderabad
Hyderabad, India
akshat.tandon@research.iiit.ac.in

Kamalakar Karlapalem
Agents and Applied Robotics Group
KCIS, IIIT-Hyderabad
Hyderabad, India
kamal@iiit.ac.in

ABSTRACT

We are given an environment with some objects (a city block area) and mobile agents moving in the environment. An agent (hider) can hide behind an object to be not seen by other agents (seekers) through their line of sight (visibility). The aim of hiders is not to be caught for the longest time, and the aim of the seekers is catch all of them in the shortest period of time. We formulate the problem by using visibility based map abstractions. Agents plan their moves by utilizing multi-armed bandits UCB reward update model. We evaluate our abstractions and strategies by simulating the game under various different scenarios.

KEYWORDS

Emergent behavior; Modelling for agent based simulation; Social Simulation

ACM Reference Format:

Akshat Tandon and Kamalakar Karlapalem. 2018. Agent Strategies for the Hide-and-Seek Game. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

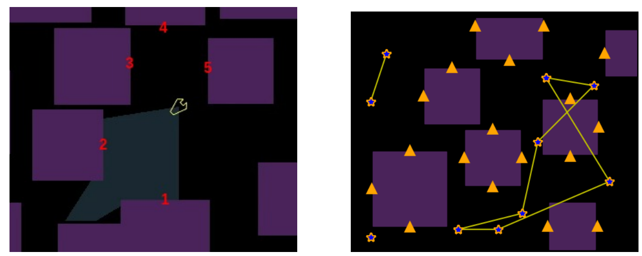
1 INTRODUCTION

Hide-and-Seek has been studied by Alpern and Gal [1], Foreman [7], Lidbetter [8], Dagan and Gal [6], Baston and Kikuta [3] for different classes of networks, multidimensional environments and agent constraints. Chapman et al. [5] have employed hide and seek game strategies to tackle cyber security problems. The focus of our model is on visibility polygons, obstacles and coverage. We delineate spatial abstractions and agent planners, which incorporate these notions. For this paper, we formalize the elements and laws of the game to make the game suitable from a 2D simulation perspective.

1.1 Players, Environment and Obstacles

The game is played in a 2D bounded, continuous, rectangular environment $\mathcal{E}$. The environment contains many obstacles $O = \{o_1, o_2, ..., o_k\}$. For simulation purposes, we consider square as a basic obstacle block and construct any arbitrary polygonal shape as a combination of contiguous squares. The game comprises of a team of hider agents $\mathcal{H} = \{h_1, h_2, ..., h_n\}$ and a team of seeker agents $\mathcal{S} = \{s_1, s_2, ..., s_m\}$. An agent is capable of taking one of the sixteen compass directional actions, each oriented at an angle of multiples of 22.5° from its base axis.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.



(a) Note the visibility region associated with the agent. Agent is only able to see strategic points 1 and 2.

(b) Triangles are $\mathcal{SP}$, stars are $\mathcal{CP}$ and the yellow lines form the contours.

Figure 1: Strategic and Coverage Points

1.2 Objectives, Elimination and Visibility

The hider team tries to maximize the elimination time of the last remaining hider in the game whereas the seeker team tries to minimize this time. The elimination time of a hider $h_k \in \mathcal{H}$ is the time at which h_k is visible to some seeker $s_l \in \mathcal{S}$. To approximate the notion of visibility in simulation, we associate a visibility polygon with each agent. A hider is visible to a seeker if the hider lies inside the seekers visibility polygon. The visibility polygon depends on the current state of an agent, changes as the agent moves and is constructed by tracing the path of uniformly spaced rays emitted from the agents current position, spread uniformly along some angle to the left and right of the agent's head facing direction.

2 SPATIAL ABSTRACTIONS AND REASONING

2.1 Strategic and Coverage Points

A strategic point is used as an abstraction for a hiding location. It is the mid point of an edge of a square obstacle. Each obstacle yields a set of strategic points and the union of these sets constitute the strategic points set $\mathcal{SP}$, of the environment. Coverage point is used as an abstraction for a seeking location. It is a point from which one or more strategic points are visible, when scanned in all the directions. A coverage point is said to cover the strategic points visible from it. An optimal set of coverage points $\mathcal{CP}$ must satisfy the following criteria:

(1) All the strategic points of the environment must be visible to at least one coverage point in the set.
(2) Maximum number of strategic points (if possible) must be visible from each coverage point in the set.

We propose an algorithm which finds such a set of optimal coverage points by utilizing an intermediary visibility graph $\mathcal{VG}$, built upon strategic points. The nodes of this graph consist of strategic points of the environment. There is an edge between any two nodes if there exists a point in $\mathcal{E}$ from which the strategic points corresponding to nodes are visible. To reduce the computational cost of constructing $\mathcal{VG}$, a discretized grid cell version $\mathcal{G}$ of environment $\mathcal{E}$ is considered. Each cell $c \in \mathcal{G}$ is represented by its center $(c_x, c_y) \in \mathcal{E}$. To compute edges of $\mathcal{VG}$, associate each strategic point with the set of grid cells visible to it, if scanned in all the directions. If the intersection of the visible cell set associated with two strategic points is not null, then there exists a point in $\mathcal{E}$ which is visible to both of these. Thus, there exists an edge between those two strategic points in $\mathcal{VG}$. An optimal set of coverage points can be obtained by (i) enumerating over all the maximal cliques (Bron and Kerbosch [4]) of the visibility graph $\mathcal{VG}$, (ii) finding the smallest set of coverage points required for covering the strategic points corresponding to the nodes of enumerated clique and (iii) taking the union of these coverage point sets. Maximal cliques of the visibility graph are considered because a maximal clique of $\mathcal{VG}$ encapsulates the set of strategic points which are visible to each other. If a set of strategic points is visible to each other, there exists a fewer number of coverage points required to cover them.

To find the smallest set of coverage points corresponding to a clique $\mathcal{CQ}$, iterate over all grid cells of $\mathcal{G}$, visible from some node (i.e. strategic point) of that clique, and find the cell whose center covers the maximum number of nodes (i.e. strategic point) of that clique. If the found cell's center covers all the strategic points of the clique, return the center as the sole member of the coverage points set, else remove the strategic points covered by the found cell from the clique $\mathcal{CQ}$ and feed the modified clique back to the algorithm to recursively find remaining coverage points.

2.2 Coverage Contours

Each coverage point is associated with a set of strategic points, which are visible to it. We present a heuristic for partitioning the set of coverage points $\mathcal{CP}$ into sets which have multiple common strategic points among themselves. Doing this enables a planner to exploit the locality around a coverage point better.

We call these partitions, coverage contours. These coverage contours serve as abstractions of traversal routes for the seeker and can be found out by utilizing a coverage graph $\mathcal{CG}$. The set of optimal coverage points constitute the set of nodes of $\mathcal{CG}$ and there is an edge between any two nodes of $\mathcal{CG}$ if the coverage points corresponding to the nodes share two or more strategic points. Each connected component of $\mathcal{CG}$ constitutes the set of coverage points of a coverage contour. To find an ordering, a pre or post order depth first traversal is performed on the connected component.

3 AGENT STRATEGIES

In coverage bandit strategy (CB), seeker agents maintain upper confidence bounds (Auer et al. [2]) over the mean rewards associated with each optimal coverage point $c_i \in \mathcal{CP}$, defined as, $UCB(c_i) = \hat{\mu}_i^t + \sqrt{\alpha \ln t / 2N_i^{t+1}}$, where α is a positive constant, t is the total number of decision epochs, $\hat{\mu}_i^t$ is the empirical mean of rewards obtained at c_i till epoch t and N_i^t is the number of times c_i has been selected till epoch t. At each decision epoch, the agent selects a coverage point which has the greatest upper confidence bound, just like in a multi armed bandit setting and then traverses the coverage contour associated with that point. It traverses the coverage contour and explores all the coverage points associated with it, updating the upper confidence bounds with the obtained rewards during the process. The agent gets a positive reward if an opponent type is detected around a coverage point and negative otherwise. In CB, instead of each seeker maintaining separate bounds, they jointly share and update a single set of confidence bounds. In strategic bandit strategy (SB), each seeker agent individually maintains bounds over strategic points. When hider agents follow SB strategy they incorporate obstructiveness and proximity. To incorporate obstructiveness, each bound of $s_i \in \mathcal{SP}$ is initialized with an obstructiveness factor, defined as, $obs(s_i) = \frac{M}{q(s_i)}$ where M is the maximum number of grid cells visible from any position and $q(s_i)$ is the number of grid cells visible from s_i, when scanned in all directions. To incorporate proximity, at each epoch, planner considers only the four nearest strategic points and selects that point amongst the four which has the highest bound. Instead of incorporating both, hider agents can either only incorporate proximity (denoted HVSB) or only incorporate obstruction (denoted HMSB).

For many different scenarios, we computed mean game completion time of 100 runs (Table 1). For seeker agents, CB outperformed SB in 10 out of 12 scenarios. For hider agents, incorporating obstruction was not always helpful since HVSB outperformed SB in 2 out of 4 scenarios. However, incorporating proximity was always advantageous since both SB and HVSB outperformed HMSB in every scenario.

$\mathcal{H} \rightarrow$	SB	HMSB	HVSB	SB	HMSB	HVSB
$\mathcal{S} \downarrow$	$\lvert\mathcal{H}\rvert = 30, \lvert\mathcal{S}\rvert = 5, \lvert\mathcal{O}\rvert = 55$			$\lvert\mathcal{H}\rvert = 50, \lvert\mathcal{S}\rvert = 5, \lvert\mathcal{O}\rvert = 110$		
CB	693.80	379.46	690.09	1175.46	783.63	1236.62
SB	939.83	441.03	652.58	1272.22	743.40	1293.67
	$\lvert\mathcal{H}\rvert = 50, \lvert\mathcal{S}\rvert = 10, \lvert\mathcal{O}\rvert = 55$			$\lvert\mathcal{H}\rvert = 70, \lvert\mathcal{S}\rvert = 10, \lvert\mathcal{O}\rvert = 110$		
CB	407.96	275.62	424.01	717.51	420.86	657.76
SB	614.73	306.59	489.42	788.14	473.70	768.99

Table 1: Mean game completion times

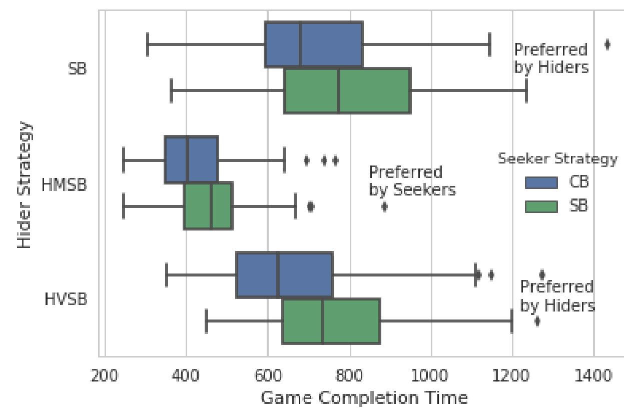


Figure 2: Completion times when $\lvert\mathcal{H}\rvert = 70$, $\lvert\mathcal{S}\rvert = 10$, $\lvert\mathcal{O}\rvert = 110$

REFERENCES

[1] Steve Alpern and Shmuel Gal. 2006. *The theory of search games and rendezvous*. Vol. 55. Springer Science & Business Media.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[3] Vic Baston and Kensaku Kikuta. 2013. Search games on networks with travelling and search costs and with arbitrary searcher starting points. *Networks* 62, 1 (2013), 72–79.

[4] Coen Bron and Joep Kerbosch. 1973. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* 16, 9 (1973), 575–577.

[5] Martin Chapman, Gareth Tyson, Peter McBurney, Michael Luck, and Simon Parsons. 2014. Playing hide-and-seek: an abstract game for cyber security. In *Proceedings of the 1st International Workshop on Agents and CyberSecurity*. ACM, 3.

[6] Arnon Dagan and Shmuel Gal. 2008. Network search games, with arbitrary searcher starting point. *Networks* 52, 3 (2008), 156–161.

[7] Joe G Foreman. 1977. Differential search games with mobile hider. *SIAM Journal on Control and Optimization* 15, 5 (1977), 841–856.

[8] Thomas Lidbetter. 2013. *Hide-and-seek and other search games*. Ph.D. Dissertation. The London School of Economics and Political Science (LSE).