Optimal Hide-and-Seek

Yvonne Andrewsen, David Morley, Adam Robertson

April 9, 2020

1 Introduction

Hide-and-seek is an engaging game in which one agent (the Hider) conceals themselves somewhere within a designated map and another agent (the Seeker) searches the map until they locate the Hider. Our optimal control problem is to identify the optimal path that the Seeker should take in order to find the Hider in minimal time.

For this project, we consider a simplified version of hide-and-seek with a two-dimensional game board and a single circular obstacle in the center. In this simplified game, the Hider is represented as a point on a grid-based map. The Hider may only appear at a finite number of possible locations. At the beginning of the game, the hider is instantiated at one of these locations randomly and stays there for the duration of the game. The Seeker, who always begins the game at the same location, is free to wander through the map until the Hider comes within the Seeker's range of visibility.

There is a single obstacle within the map that obstructs the Seeker's view of a portion of the game board. The Seeker must navigate around this obstacle in order to see points that, at the beginning of the game, are on the opposite side of the Seeker's starting point. Feasible curves are paths through the grid that the Seeker can take while they look for the Hider.

Since the Hider is guaranteed to exist on only a finite number of points, we consider this problem, not as finding the hider's location in one specific game, but as finding the best path to view all possible hiding places as quickly as possible. Furthermore, the Seeker's velocity is constrained based on their position in the game board. As the Seeker approaches the boundary of the obstacle, their maximum possible velocity slows. This slowing discourages the Seeker from coming too close to the obstacle, simulating a real life environment where it hurts the Seeker to run into the obstacle.

2 Background Research

In 2014, Alpern and Lidbetter [1] published research studying Hide-and-Seek that treated the game board as a finite network with an immobile Hider and a mobile Searcher. Their strategy divides the infinite set of possible paths into a tree with a finite number of representations of classes of paths that the Searcher can take. Their algorithm relies on searching high-density regions first, then moving to low-density regions if the Hider is not found.

Studies on the Optimal Search Plan [2] is a book that provides a longitudinal overview of the various cases of search problems and the work done to solve them. The case that we are considering falls under the category of a stationary target in a discrete target space with a continuous search path. This book is a valuable resource for discovering other research approaches and variations of the problem that we are considering.

3 Problem Setup

3.1 Notation

We consider the game board B as a subset of $\mathbb{R}^2$, defined as

$$B = [-2, 2] \times [-2, 2],$$

The obstacle that the car must navigate around is a circle, centered at the origin, of radius 1. Let $O = (x, y) : x^2 + y^2 <= 1$ be the obstacle.

The Seeker always begins at the point (-2,-2) and is free to travel on any continuous path within $B \setminus O$. The Hider, however, is constrained to appear only at points with integer coordinates. Hence, there are a discrete, finite number of points where the Hider may exist. Once the Seeker has "viewed" all of these points, they are guaranteed to have located the Hider. We consider a hiding point as "viewed" when the straight line that connects the Seeker to the hiding point is unimpeded by the obstacle. Let t_f be the first instant at which the Seeker has viewed all possible hiding points, thereby winning the game.

The Seeker's maximum possible velocity is constrained by a function that depends on the Seeker's distance from the center of the obstacle. Let

$$v(x, y) = e^{-\frac{1}{x^2+y^2}}$$

be the function that describes the Seeker's velocity.

Because the game board is symmetric about the line $y = x$, we assume (without loss of generality) that the Seeker departs from their initial position in a counter-clockwise direction. The results from this analysis would also apply in the case that the car moves clockwise, with the optimal curve being a reflection about $y = x$ of the optimal curve in the counter-clockwise case. The final time t_f is the same in both cases.

See **Figure 1** for a visualization of the game board.

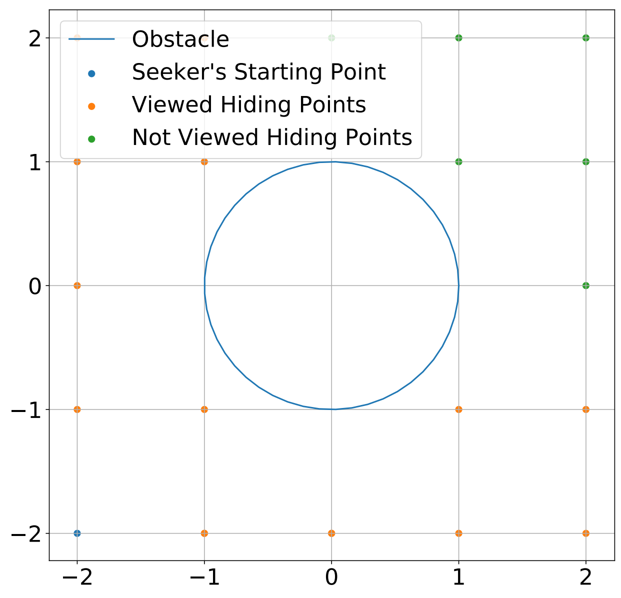

Figure 1: A plot of the Game Board at the beginning of the game. Orange points are places visible to the car at the start of the game.

3.2 A Clever Observation

Observe that when the Seeker crosses the line $y = 2 - \sqrt{3}x$, then they are guaranteed to have viewed all possible hiding points.

See **Figure 2** for a visualization of several possible solutions to the hide-and-seek problem.

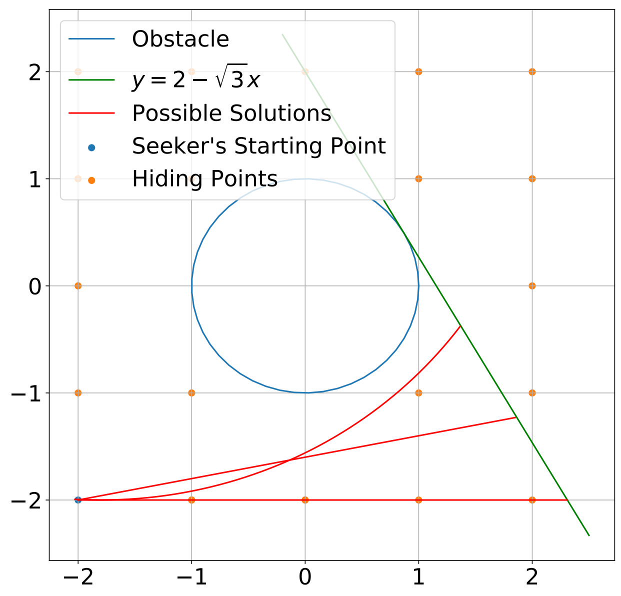

Figure 2: A plot of possible curves for the Seeker to follow.

4 Finding an ODE to Solve

We seek to minimize

$$J(\theta) = \int_{t_0}^{t_f} 1 dt$$

$$\text{SUBJECT TO } \dot{\mathbf{z}}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \mathrm{e}^{-\frac{1}{x^2+y^2}} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} = \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} = f(\mathbf{z}, \theta)$$

$$\mathbf{z}(t_0) = \begin{pmatrix} z_1(t_0) \\ z_2(t_0) \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \qquad \begin{pmatrix} z_1(t_f) \\ z_2(t_f) \end{pmatrix} \in S$$

Here, θ is our control and S is the line $y = 2 - \sqrt{3}x$.

The Hamiltonian in this case is

$$\begin{aligned} H =& \langle p, f(\mathbf{z}, \theta) \rangle - \mathcal{L} \\ =& \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \begin{pmatrix} \cos(\theta) & \sin(\theta) \end{pmatrix} \begin{pmatrix} p_1(t) \\ p_2(t) \end{pmatrix} - 1 \\ =& \mathrm{e}^{-\|\mathbf{z}\|^{-2}} (p_1(t)\cos(\theta) + p_2(t)\sin(\theta)) - 1 \end{aligned}$$

Furthermore,

$$-H_z = \dot{p} = \begin{pmatrix} \dot{p_1} \\ \dot{p_2} \end{pmatrix} = - \begin{pmatrix} \dfrac{2z_1 \mathrm{e}^{-\|\mathbf{z}\|^{-2}}}{\|\mathbf{z}\|^4} \\ \dfrac{2z_2 \mathrm{e}^{-\|\mathbf{z}\|^{-2}}}{\|\mathbf{z}\|^4} \end{pmatrix} (p_1 \cos(\theta) + p_2 \sin(\theta))$$

Using Pontryagin's Maximum Principle for the Basic Variable-Endpoint Control Problem, properties 2-4 (see Liberzon, page 104), we have:

(Property 1)

$$\begin{aligned} \mathbf{z}(t_0) =& (-2, 2)^T \\ \mathbf{z}(t_f) \in& S \end{aligned}$$

(Property 2)

$$\begin{aligned} 0 = H_\theta =& \mathrm{e}^{-\|\mathbf{z}\|^{-2}} (-p_1(t)\sin(\theta) + p_2(t)\cos(\theta)) \\ \Rightarrow& 0 = -p_1(t)\sin(\theta) + p_2(t)\cos(\theta) \\ \Rightarrow& p_1(t)\sin(\theta) = p_2(t)\cos(\theta) \\ \Rightarrow& \frac{\sin(\theta)}{\cos(\theta)} = \tan(\theta) = \frac{p_2(t)}{p_1(t)} \\ \Rightarrow& \theta = \arctan\left(\frac{p_2(t)}{p_1(t)}\right) \end{aligned}$$

(Property 3)

$$\begin{aligned} 0 =& H = \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \left(p_1(t)\cos(\theta) + p_2(t)\sin(\theta)\right) - 1 \\ =& e^{-\|\mathbf{z}\|^{-2}} \sqrt{p_1^2 + p_2^2} - 1 \\ \Rightarrow \frac{1}{e^{-\|\mathbf{z}\|^{-2}}} =& \sqrt{p_1^2 + p_2^2} \\ \Rightarrow \left(e^{\|\mathbf{z}\|^{-2}}\right)^2 =& p_1^2 + p_2^2 \\ \Rightarrow 0 =& p_1^2 + p_2^2 - e^{2\|\mathbf{z}\|^{-2}} \end{aligned}$$

(Property 4)
$\langle p(t_f), d\rangle = 0 \quad \forall d \in T_{\mathbf{z}(t_f)}S$ where $T_{\mathbf{z}(t_f)}S = \{d \in \mathbb{R}^2 : \langle \Delta h(\mathbf{z}(t_f)), d\rangle = 0\}$. In this case $h = \sqrt{3}z_1 + z_2 - 2$ which implies that $\Delta h = \begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix}$ and $d = \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}$. Thus plugging back into (4) it is clear that $p_1(t_f) = \sqrt{3}p_2(t_f)$ which gives us that $\theta(t_f) = \frac{\pi}{3}$.

Putting all these facts together gives us the following system that we can solve numerically:

$$\begin{aligned} \dot{\mathbf{z}}(t) =& f(\mathbf{z}, p) = \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \begin{pmatrix} \frac{p_1}{\sqrt{p_1^2+p_2^2}} \\ \frac{p_2}{\sqrt{p_1^2+p_2^2}} \end{pmatrix} = \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \|p\|^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \\ \dot{p}(t) =& g(\mathbf{z}, p) = -\begin{pmatrix} 2z_1 \|\mathbf{z}\|^{-4} \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \\ 2z_2 \|\mathbf{z}\|^{-4} \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \end{pmatrix} \sqrt{p_1^2 + p_2^2} = -2\|p\| \|\mathbf{z}\|^{-4} \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\ \mathbf{z}(t_0) =& (-2, -2)^T \\ z_2(t_f) =& 2 - \sqrt{3}z_1(t_f) \\ 0 =& p(t_f) \cdot \begin{pmatrix} -1 & \sqrt{3} \end{pmatrix}^T \end{aligned}$$

5 Numerical Method

Combining the derivations of the last section, we set up a system that can be plugged into a numerical solver. We chose to use the python package `scipy.integrate.solve_bvp` to solve our system. However, because our problem has an open end time, and `solve_bvp` requires a fixed end time, we perform a change of variables in order to fix the end state. We define, t_f to be a constant. Now we have,

$$\begin{aligned} t =& st_f, \quad s \in [0, 1] \\ \frac{d}{dt} =& \frac{d}{ds}\frac{ds}{dt} \end{aligned}$$

We note that $s = \frac{t}{t_f}$ which implies that $\frac{ds}{dt} = \frac{1}{t_f}$. Plugging this in gives,

$$\begin{aligned}\frac{d}{dt} =& \frac{d}{ds}\frac{1}{tf} \\ \Rightarrow& \frac{d}{ds} = t_f \frac{d}{dt}\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{d}{ds}\mathbf{z} =& t_f \frac{d}{dt}\mathbf{z} = t_f \dot{\mathbf{z}} = t_f \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \|p\|^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \\ \frac{d}{ds}p =& t_f \frac{d}{dt}p = t_f \dot{p} = -2t_f \|p\| \|\mathbf{z}\|^{-4} \mathrm{e}^{-\|\mathbf{z}\|^{-2}} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}\end{aligned}$$

This gives rise to the following fixed endpoint system:

$$\begin{aligned} J(\theta) =& \int_0^1 1 ds \\ \text{WITH BOUNDARY CONDITIONS } \mathbf{z}(t_0) =& \begin{pmatrix} -2 \\ -2 \end{pmatrix} \\ 0 =& 2 - \sqrt(3) z_1(t_f) - z_2(t_f) \\ 0 =& p_1(t_f) - \sqrt{3} p_2(t_f) \\ 0 =& p_1(t_0)^2 + p_2(t_0)^2 - \mathrm{e}^{\frac{1}{4}} \end{aligned}$$

6 Solution Interpretation

The numerical approximation method did not perform as had expected. See **Figure 3** for a plot of our solution.

Our modeling choice to punish the Seeker for moving too close to the boundary of the obstacle, instead of explicitly preventing it from permeating the obstacle, did not work. The solver shows that the optimal path that minimizes time is to pass through the circle. This indicates that the penalty was not sufficiently high to force the Seeker to stay out of the obstacle. A plot of the velocity function $v(\mathbf{z}) = e^{-||\mathbf{z}||^{-2}}$ is shown in **Figure 4**. This function drastically reduces velocity as the Seeker approaches the origin, but the velocity penalty quickly tapers as the Seeker moves a short distance from the origin.

7 Further Work

Much still remains to be done in finding a perfect solution to this variation of the Hide and Seek problem. Since the main error in our solution came because of a poorly chosen velocity function, future efforts could be spent in finding a more suitable velocity function. One possible change could be to define the velocity function in such a way as to incentivize

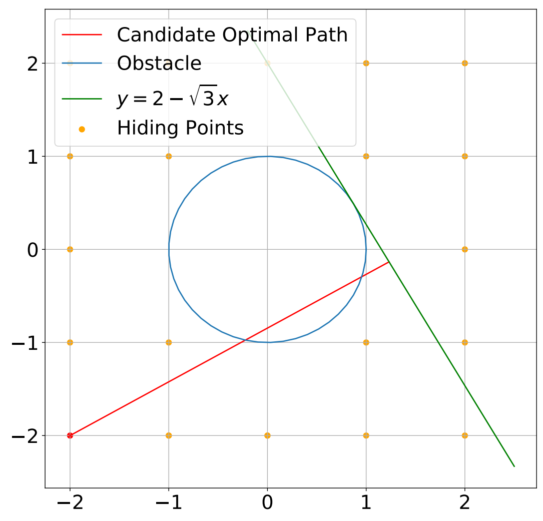

Figure 3: A plot of the result of our numerical method.

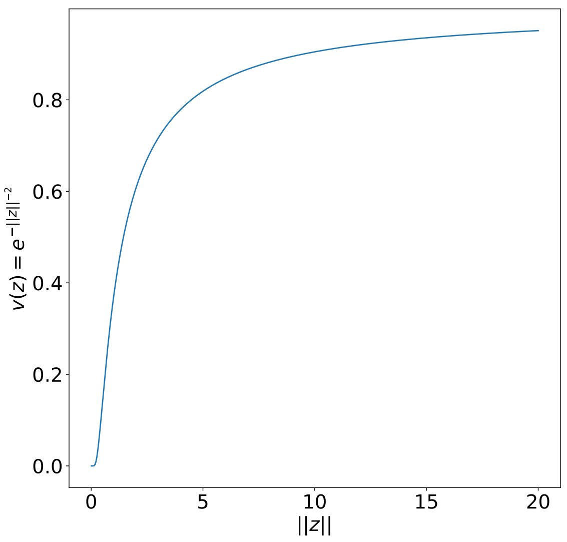

Figure 4: A plot of the velocity function.

staying out of the obstacle. We implemented a variation of this idea that made velocity increase proportional to the radius. See **Figure 5** for our initial attempt at this method.

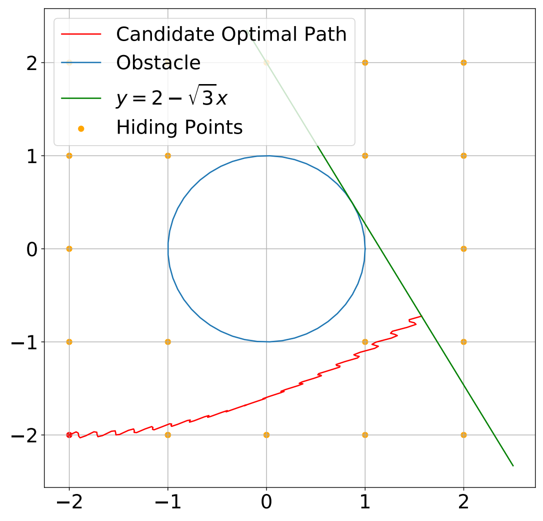

Figure 5: Another potential solution for the motion of the Seeker.

Other approaches could consider different solution paths. For example, it might be possible to formulate the problem like the River Crossing problem and use spectral methods in the numerical solution.

8 Conclusion

The optimal hide-and-seek problem is an interesting problem. We were able to design a simplified version of the problem with a unique velocity constraint. This version of the game could have many real-world applications, such as the problem of a farmer driving a tractor around a silo. The tractor may slow down while turning in order to avoid damaging the silo. A method similar to what we used here could find the optimal path for the farmer to take.

Even though we were unable to find a candidate optimal path with reasonable behavior, we have learned a great deal about Pontryagin's Maximum Principle and about constructing a problem with constraints that require it to have an optimal control that is not immediately obvious, forming a system of ODEs to find a candidate optimal control, and writing a numerical method to attempt to solve that system.

References

[1] S. Alpern. Searching a variable speed network. *Mathematics of Operations Research*, 39(3), 2015. Available at https://doi.org/10.1287/moor.2013.0634.

[2] K. Iida. *Studies on the Optimal Search Plan.* 1992. Available at https://www.springer.com/gp/book/9780387977393aboutBook.