Network Search Games with Asymmetric Travel Times

Steve Alpern
Department of Mathematics, London School of Economics
Houghton Street, London WC2A 2AE, United Kingdom
s.alpern@lse.ac.uk
Abstract

A point $H$ is hidden in a rooted tree $Q$ which is endowed with asymmetric distances (travel times) between nodes. We determine the randomized search strategy, starting from the root, which minimizes the expected time to reach $H$, in the worst case. This is equivalent to a zero-sum search game $\Gamma'(Q)$, with minimizing Searcher, maximizing Hider and payoff equal to the capture time. The worst Hiding distribution (over the leaves) from the Searcher’s viewpoint, is one where at every node $i$ the probability of each branch is proportional to the minimum time required to tour it from $i$. The optimal randomized search is a mixture over depth-first searches. We also consider briefly some other networks and the possibility of a mobile Hider.

Our formulation with asymmetric travel times generalizes that of Gal (1979) for symmetric travel times and also the search games of Kikuta (1995) and Kikuta and Ruckle (1994), who posited search costs $c_i$ at each node $i$ which were added to the travel time to obtain the payoff.

We also briefly consider what happens if we allow the Searcher (Hider) to start (hide) at any leaf node. We determine when properties found by Dagan and Gal (2008) for the symmetric version of such games hold in our asymmetric context.

Keywords: search game, tree, asymmetric distance, Chinese Postman path
AMS Subject Classifications: 90B40, 91A25, 90B15
Abreviated Title: Network Search Games
1 Introduction

We consider the problem faced by a Searcher who has to find a point $H$ hidden on a rooted tree $Q$, which has given travel times $F_a$ and $R_a$ to traverse each arc $a$ in the forwards (away from the root) and reverse directions. We determine the randomized search strategy, starting from the root $O$, which minimizes the expected time to find $H$, in the worst case. Our approach is to solve the the zero-sum search game $\Gamma$ in which the Searcher is the minimizer, the Hider who picks $H$ is the maximizer and the payoff is the expected time $T$ taken for the Searcher to reach the point $H$. Such games, in the distance-symmetric form, were first posed by Rufus Isaacs in his 1965 book Differential Games [13]. Clearly any hiding place $H$ which is not a leaf node is dominated by points further from the root, so we can assume $H$ is a leaf. To simplify the presentation we assume $Q$ is a binary tree, as we can reduce any tree to a strategically equivalent binary tree.

A typical tree $Q$ is illustrated on the left of Figure 1, using the convention that the forward and reverse travel times are respectively placed to the left and right of the arc. The optimal strategies for the players are best described by branching functions, functions that assign probability distributions to the forward arcs at every non-leaf node. The optimal branching functions $r$ for the Searcher and $p$ for the Hider are drawn in the middle and right-hand pictures. Given any branching function $q$, the branching strategy $B(q)$ is described as follows:

1. Leaving a node the first time, take forward arc $a$ with probability $q(a)$.
2. Leaving that node next time, take the untraversed forward arc.
3. Leaving that node for the third time, take the reverse arc.

For the Searcher, the optimal strategy is to follow $B(r)$ until reaching $H$. For the Hider, the optimal strategy is to follow $B(p)$ until a leaf node is reached, and to hide there. The corresponding optimal Hider distribution $h$, considered as a probability measure on $Q$, has probability of hiding at a leaf node $w$ given in closed form as

$$h(w) = \prod_{O < a < w} p(a) ,$$

where the product is taken over the arcs in the unique path from root $O$ to leaf $w$. The picture on the right gives $h(L), h(M)$ and $h(R)$ in parentheses. The optimal Hider branching probability $p(a)$ for a forward arc $a$ out of any
node $y$ is given by the time taken to tour the branch from $a$ divided by the time taken to tour the subtree rooted at $y$ (with both tours taken from $y$). For example, the left arc out of node $x$ has probability $10/15 = 2/3$ where 10 and 15 are respectively the times taken to tour the left branch and the full subtree starting at $x$. For reasons discussed later, we call this optimal distribution $\hat{h}$ the Equal Branch Density (EBD) distribution. For the Searcher we will need a recursion formula to calculate the optimal branching function $r$, but we note for the present that all the sample paths (or pure strategies) resulting from $B(r)$ are depth-first searches.

Our asymmetric model generalizes the symmetric model ($F_a = R_a$ for all arcs $a$) of Gal (1979). For symmetric trees, he showed that the value of the search game was the total length of the tree, and thus independent of the location of the root. This root independence fails in our model, even for the simplest single arc tree, where the value is $F_a$, but would be $R_a$ if the root were moved to the other node. Gal also found that an equiprobable mixture of a Chinese Postman Tour and it’s time reversed tour ensures that $T_o = 2$; where $o$ is the minimum time to tour $Q$ from $O$, and that $\tau_o/2$ is the value. When travel time symmetry is relaxed, this is no longer true, but we do have (see (5)) the estimate $V \leq (1/2) \left( \tau_o + \sum_a |F_a - R_a| \right)$.

Our model is also related to the search games on symmetric trees formulated by Kikuta and Ruckle [16] and Kikuta [15], where each node $i$ was assigned an additional search cost $c_i$ representing the time needed to search out that node (or that node represents another graph). On arriving at a node, the Searcher could simply pass through it without incurring this cost, or carry out a search. In the latter case the cost $c_i$ is incurred, whether or not the Hider is there. More generally, each node $i$ can have compound search costs $(m_i, t_i)$. If the node is searched and the Hider is there, he is found in mean time $m_i$. If not, the time taken to ascertain he is not there is $t_i$, with $m_i \leq t_i$. Such search costs at nodes $i$ can be incorporated into an asymmetric distance model by replacing them with additional leaf arcs $a$ at node $i$ with travel times $F_a = m_i$ and $R_a = t_i - m_i$. (If the arc is searched and the Hider was at the corresponding node, the additional time is $m_i$; if the Hider was not there, the additional time is $F_a + R_a = t_i$.) Our recursive formulae, suitably interpreted, are the same as those derived by Kikuta and Ruckle [16]. (This has to be so, as their costs can be viewed as travel times in our model.)

We note that other familiar problems like the Travelling Salesman and Chinese Postman have also been subsequently studied in distance-asymmetric, or ‘windy’, versions [18]. This is the first attempt to introduce asymmetric distances to network search games.

Following their formulation by Isaacs [13], search games on networks have been extensively studied. Gal [11] extended his analysis of trees to a wider class of networks called weakly Eulerian networks, following an important intermediate result of Reinierse and Potter [19]. Interesting networks not in this class have been studied by Jotshi and Batta [14] and Pavlovic [17]. The existence of the value of search games on all networks follows from Alpern and Gal [5]. Papers studying the variation in which the Hider is allowed to move include [20],[1],[4] and [8]. The monographs on the area of search games are Alpern and Gal [5] and Garnaev [12].

The paper is organized as follows. Section 2 defines the notions of search density and depth-first paths, and shows why depth-first paths are optimal
responses to the EBD Hider distribution. Section 3 presents our solution to search games on trees without symmetric travel times and explains the connections with previous models. Section 4 considers various extensions to our model. First, in Sections 4.1 to 4.3 we relax the requirement that $Q$ is a tree, studying a circle and then a lollipop (circle with a ray) network. That latter exhibits a new property, namely that the searcher must explore a subgraph (specifically, the circle) differently, depending on when it is searched (i.e. before or after the ray). Finally, in Section 4.4 we relax the assumption that the Hider cannot move. We show that in the case where $Q$ is a circle the solution changes only in a superficial way from the solution given for a symmetric circle by Zeliken [20]. In Section 5 we return to trees (in fact stars), but assume that the Hider and Searcher can hide and start, respectively, at any leaf node. We derive an indicator function $I$ of the travel times such that if $I \geq 0$ the qualitative behaviour shown by Dagan and Gal [9] for symmetric trees remains optimal, that the Hider uses an EBD distribution, taking the root to be the ‘center’ of the tree. (We modify the notion of center asymmetrically. The other property found by Dagan and Gal, that the Searcher can start and end at diametrical nodes, fails generically.

We use throughout the usual conventions that upper case letters $S, H$ refer to Searcher and Hider pure strategies and that lower case letters $s, h$ refer to mixed strategies. Also, we let $T$ be the capture time when its arguments are pure and expected capture time when they are mixed.
2 Search Density and Depth-First Paths

In this section we consider the subproblem of how to search a network to minimize the expected time to find a Hider whose distribution \( h \) is known. In this setting it is sometimes useful to compare the searches of probabilistically disjoint regions starting and ending at a common point \( x \). The search density \( \rho \) of such a region \( A \) is simply the probability that the Hider is in it, divided by the time taken to search it.

We establish two simple results, the Search Density Lemma and the Depth-First Lemma, which will be used to solve the search game \( \Gamma \).

2.1 Search Density Lemma

A simple observation (used earlier by Pavlovic [17], extended by Alpern and Howard ([7])) and applied to rendezvous search by Alpern and Beck ([2],[3])) is that, optimally, regions of higher search density should be searched first and regions of equal search density can be searched consecutively. This concept is implicitly or explicitly used in much of the earlier literature, but for completeness we formally establish a simple case here.

Fix a network \( Q \) and a Hider distribution \( h \) on \( Q \). If \( S(t) \) is a Searcher path with cumulative capture distribution \( G(x) = \Pr(T(S,H) \leq x) = h(S[0,x]) \), then the expected capture time \( T \) is given by \( T(S,h) = \int_0^\infty t \, dG(t) \). The following result applies to general search problems, not just to trees.

Lemma 1 (Search Density) Fix a network \( Q \) and a Hider distribution \( h \). Suppose that \( a < b < c \), \( S(a) = S(b) = S(c) \) and that \( S \) searches probabilistically disjoint regions \( A \) and \( B \) of \( Q \) in time intervals \([a,b]\) and \([b,c]\), that is, \( h(S[a,c]) = h(S[a,b]) + h(S[b,c]) \). Let \( S' \) be the same path as \( S \) except that it searches \( A \) and \( B \) in the other order (\( B \) during \([a,a+(c-b)]\) and \( A \) during \([a+(c-b),c]\))

\[
\text{If } \rho(B) = \frac{G(c) - G(b)}{c - b} \geq \rho(A) = \frac{G(b) - G(a)}{b - a}, \text{ then } T(S',h) \leq T(S,h).
\]

In other words the search with higher search density should be carried out first. Furthermore if \( S \) searches three probabilistically disjoint regions \( A, B, C \) in that order, where \( A \) and \( C \) have the same search density, then \( A \) and \( C \) can be searched consecutively without increasing the expected search time.

Proof. For the first part, note that the search \( S' \) reaches points in \( A \) in time \( c - b \) later than does \( S \), and points in \( B \) in time \( b - a \) earlier. So the difference in search times, \( T(S,h) - T(S',h) \), is given by

\[
\int_a^c t \, dG(t) - \left[ \int_a^b (t + (c - b)) \, dG(t) + \int_b^c (t - (b - a)) \, dG(t) \right]
\]

\[
= \int_b^c (b-a) \, dG(t) - \int_a^b (c-b) \, dG(t)
\]

\[
= (b-a)(G(c) - G(b)) - (c-b)(G(b) - G(a))
\]

\[
= (b-a)(c-b)(\rho(B) - \rho(A)) \geq 0.
\]
For the second part suppose note that since \( \rho(B) \leq \rho(A) \) or \( \rho(B) \geq \rho(C) \), we can use the first part to change the search order of either \( A \) and \( B \), or of \( B \) and \( C \), without increasing \( T \). Either change searches \( A \) and \( C \) consecutively, as claimed.

**2.2 Depth-First Paths**

Here we show that there is an optimal response to the EBD Hider distribution which is a depth-first search. We begin by grouping together several definitions and notations relating to trees.

**Definition 2** Every arc or node \( x \) of \( Q \) defines a subtree \( \sigma(x) \) consisting of all points of \( Q \) connected to the root via \( x \). If \( x \) is a node, it is the root of \( \sigma(x) \); if \( x \) is an arc, its base node (towards \( O \)) is the root. The value of the search game on the tree \( \sigma(x) \) is denoted by \( v_x \). The total time required to tour \( \sigma(x) \) from its base (the sum of the forward and reverse travel times on its arcs) is its search time, denoted by \( \tau_x \). Its search density \( \rho \) is defined by \( \rho(\sigma(x)) = h(\sigma(x))/\tau_x \).

The EBD distribution \( \hat{h} \) on a rooted tree, that we earlier defined recursively (1), can be defined in terms of search density \( \rho \) as the unique distribution which gives equal search density to all the branches at a node. It follows from the Search Density Lemma that there is an optimal search path \( S \) against \( \hat{h} \), which, upon reaching any node \( x \), tours all of its branches consecutively. Such a path is called depth-first. This is equivalent to saying that \( S \) leaves a node via an untraversed forward arc, whenever it is possible (if not it takes the unique reverse arc). If a search path searches the branches at a node consecutively, for all nodes of level up to \( k \) (and not for some level \( k + 1 \) node) we say it is branch-consecutive of level \( k \). So a path is depth-first if it is branch-consecutive up to the level of the root node. We now show that depth-first searches should be used against the EBD hiding strategy.

**Lemma 3 (Depth-First)** Suppose the Hider adopts the EBD strategy \( \hat{h} \) on a tree \( Q, O \). Then there is an optimal Searcher response strategy \( S \) which is depth-first.

**Proof.** Suppose on the contrary that there is a weighted network \( Q, O \) for which optimal Searcher strategies can be branch-consecutive only up to some level \( k \) which is less than the level of \( O \). Among these, let \( \hat{S} \) be one with the fewest nodes of level \( k + 1 \) at which it is not branch-consecutive, and let \( j \) be one such node. Then for any forward arc \( a \) at \( j \), \( \hat{S} \) tours from \( j \) the branch (subtree) \( \sigma(a) \) exhaustively in time \( \tau_a \) (since its first branch node has level \( k \)). However there must be two such arcs \( a \) and \( b \) at \( j \) such that \( \hat{S} \) does not search \( \sigma(a) \) and \( \sigma(b) \) consecutively. Since by definition of \( \hat{h} \) the two branches \( \sigma(a) \) and \( \sigma(b) \) have the same search density, the Search Density Lemma asserts that the searches of \( \sigma(a) \) and \( \sigma(b) \) can be moved together (either both before or both after the intervening search which is edge-disjoint from \( \sigma(j) \)) without increasing the expected search time. Repeat this process for all the branches at \( j \), calling the resulting path \( S^* \). But \( S^* \) contradicts the assumed maximality properties of \( \hat{S} \). ■
3 Recursion Method and Main Result

In this section we state and justify the recursion formulae which enable us to solve the game $\Gamma$. The idea is that we recursively attach labels $[v_x, \tau_x]$ to all nodes and arcs of $Q$, starting with the trivial labels $[0, 0]$ on the leaf nodes, and working backwards. In a similar manner we determine the Searcher and Hider branching functions $r$ and $p$. The determination of the numbers $\tau_x$ and $p$ can in fact be done directly, without recursion. The analysis of the game on the tree of Figure 1, is shown below and will be discussed in the next section.

The method by which we attach labels to arcs based on the label on their forward node is trivial, but for completeness we state it as follows. The tree \( \sigma (a) \) begins with a single arc $a$ so there is no choice for the Searcher but to go to its forward end $z$ and play optimally on $z$.

**Lemma 4** Let $z$ be the forward node of an arc $a$. Suppose that the EBD distribution is uniquely optimal for the Hider on the network $\sigma(z)$, and that the labels on $z$ (the value and search time of $\sigma(z)$) are $v_z$ and $\tau_z$. Then the labels on the arc $a$ are

\[ v_a = F_a + v_z, \quad \tau_a = \tau_z + F_a + R_a. \]  

(2)

This is illustrated by the labels of the arcs in the middle picture of Figure 2. A more complicated element of the recursion occurs when labelling nodes are a function of the labels of their forward arcs. The following result can be read first under the assumption that $Q$ is a binary tree, where there are exactly two forward arcs $a$ and $b$ at the node $y$, and $\sigma(b)$ has the usual interpretation.

We use the formulation below only for the sake of generality, as the analysis of binary trees would in fact be sufficient.

**Lemma 5 (Recursion)** Let $Q, y$ be a rooted binary tree with the property that the EBD distribution is optimal for the Hider for the games on its proper subtrees. Then the EBD distribution is also optimal on $Q, y$. If $a$ is a forward arc at node $y$, let $\sigma(a)$ denote the branch at $a$ and let $\sigma(b)$ denote the complementary subtree $\sigma(y) - \sigma(a)$. If $[v_a, \tau_a]$ and $[v_b, \tau_b]$ are the value-searchtime labels on the subtrees $\sigma(a)$ and $\sigma(b)$, then the label $[v_y, \tau_y]$ on $y$ (the Value and search times of $Q, y$) are given by

\[ v_y = \frac{\tau_a v_a + \tau_b v_b + \tau_a \tau_b}{\tau_a + \tau_b}, \quad \tau_y = \tau_a + \tau_b. \]  

(3)
An optimal strategy for the Searcher is the branching strategy $B(r)$, where the probability $r_a$ of taking arc $a$ when first leaving $y$ is given by
\[ r_a = \frac{\tau_b + v_a - v_b}{\tau_a + \tau_b} = \frac{\tau_b + v_a - v_b}{\tau_y} \].

**Proof.** Assume the Hider adopts the EBD distribution $\bar{h}$ on $Q$, so that he hides in $\sigma(a)$ with probability $p_a = \tau_a / (\tau_a + \tau_b)$. By the Depth-First Lemma, the Searcher can optimally respond with a depth first strategy $S$, which we can assume (by renaming) first searches subtree $\sigma(a)$ and then searches $\sigma(b)$. Then $V(Q) = v_y \geq T(S, \bar{h})$.

\[ p_a v_a + (1 - p_a) (\tau_a + v_b) = \frac{\tau_a}{\tau_a + \tau_b} (v_a) + \frac{\tau_b}{\tau_a + \tau_b} (\tau_a + v_b) = \frac{\tau_a v_a + \tau_b v_b + \tau_a \tau_b}{\tau_a + \tau_b} \].

(The symmetry of the bound in terms of $a$ and $b$ shows that the order of the search is irrelevant.)

Next suppose that the Searcher follows the random depth first strategy $RDF(r)$, searching $\sigma(a)$ first with the stated probability $r_a$. Let $q$ denote the probability the Hider is in $\sigma(a)$. Then $T \leq E(q)$, where the expected capture time $E(q)$ satisfies
\[
E(q) \leq r_a (q (v_a) + (1 - q) (\tau_a + v_b)) + (1 - r_a) (q (\tau_b + v_a) + (1 - q) (v_b)) \\
= \frac{\tau_b + v_a - v_b}{\tau_a + \tau_b} (q (v_a) + (1 - q) (\tau_a + v_b)) \\
+ \frac{\tau_a + v_b - v_a}{\tau_a + \tau_b} (q (\tau_b + v_a) + (1 - q) (v_b)) \\
= \frac{\tau_a v_a + \tau_b v_b + \tau_a \tau_b}{\tau_a + \tau_b}, \text{ independently of } q, \text{ so we are done.}
\]

The uniqueness of the optimal values for $p_a$ and $r_a$ follow from the fact that the following zero sum game has a unique solution:

<table>
<thead>
<tr>
<th></th>
<th>Search $\sigma(a)$ first</th>
<th>Search $\sigma(b)$ first</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide in $\sigma(a)$</td>
<td>$v_a$</td>
<td>$\tau_b + v_a$</td>
</tr>
<tr>
<td>Hide in $\sigma(b)$</td>
<td>$\tau_a + v_b$</td>
<td>$v_b$</td>
</tr>
</tbody>
</table>

Since the EBD distribution is trivially optimal for trees with a single arc, it follows from the Recursion Lemma that

**Theorem 6** For the search game $\Gamma$ on a rooted tree $Q, O$,

1. The EBD distribution is uniquely optimal for the Hider.

2. The branching strategy $B(r)$, where the Searcher’s branching function $r$ is calculated by (4), is optimal for the Hider. On a binary tree, this branching function is the only one for which $B(r)$ is optimal.

3. The Value $V$ can also be calculated recursively by (2) and (3).

We note that if there is non-binary branching at a node $y$, then there is a choice to be made in choosing the arc $a$ used in the Recursion Lemma. This does
not affect the value assigned to \( y \) because the value formula (3) is associative, and the order in which the choices are grouped doesn’t matter. However it does affect the mixed strategy which corresponds to the resulting branching strategy. Thus we can find different optimal mixed strategies by converting the tree to a binary tree in different ways. If we don’t wish to partition multiple branches into \( \sigma (a) \) and its complement, as done in the Recursion Lemma, we can use the following formula for \( v_y \) as a function of the labels \([v_i, \tau_i]\) on its forward arcs \( a_j, j = 1, \ldots, k\):

\[
v_y = \frac{1}{y} \left( \sum_{j=1}^{k} v_j \tau_j + \sum_{j \neq l} \tau_j \tau_l \right).
\]

### 3.1 Illustration of Recursion Method

We illustrate the use of the recursion formulae for the \( v_x \) (3) and the \( r_a \) (4) for the tree of Figure 1. The recursive calculation of the \( v_x \) starting at leaf nodes \( x \) with \( v_x = 0 \), down to leaf arcs, and to further nodes and arcs, is drawn in the middle picture of Figure 2 (along with the easily computed numbers \( x \)). Once the numbers \([v_a, \tau_a]\) and \([v_b, \tau_b]\) are known for the forward arcs at a node, the Searcher’s branching probabilities \( r_a \) and \( r_b \) (of taking \( a \) or \( b \), if neither has been taken) is given by (4), as drawn on the right. The branching strategy \( B(r) \) is equivalent to the mixed strategy in which the depth-first searches LMR, MLR, RLM and RML are carried out with respective probabilities \( \frac{1}{3} \frac{1}{5} = \frac{1}{15} \), \( \frac{1}{3} \frac{4}{5} = \frac{4}{15} \), \( \frac{2}{3} \frac{1}{5} = \frac{8}{15} \) and \( \frac{2}{3} \frac{1}{5} = \frac{2}{15} \). The branching function \( r \) is the uniquely optimal one, but there are other optimal mixed strategies.

### 3.2 Measuring Asymmetry

Before seeing how our results on (asymmetric) trees reduce to Gal’s results on symmetric trees, we first consider what happens if \( Q, O \) is close to being symmetric. We define the asymmetry number \( \alpha = \alpha (Q, O) = \sum_{\alpha \in A} |F_\alpha - R_\alpha| \), which is 0 for symmetric trees and is a measure of the degree of asymmetry. It is interesting to check how well Gal’s randomized Chinese postman paths (an equiprobable mixture \( s \) of a minimal tour \( S_1 \) of \( Q \) from \( O \) and its reverse path \( S_2(t) \)) do on our (asymmetric) trees. Fix any leaf node \( x \) and note that in the sum \( T(s, x) = T(S_1, x) + T(S_2, x) \), every arc appears either once in each direction (call the set of such arcs \( A_1 \)) or twice in its forward direction (call the set of such arcs \( A_2 \)). Thus (with \( A \) the set of all arcs and \( \tau = \tau_o \) the search time of \( Q \), we have

\[
2T(s, x) - \tau = (T(S_1, x) + T(S_2, x)) - \tau
\]

\[
= \sum_{a \in A_1} (F_a + R_a) + \sum_{a \in A_2} 2F_a - \sum_{a \in A} (F_a + R_a) = \sum_{a \in A_2} F_a - R_a \leq \alpha.
\]

So \( T(s, x) \leq \frac{\tau + \alpha}{2} \), and hence \( V = V(Q, O) \leq \frac{\tau + \alpha}{2} \).

### 3.3 Symmetric Trees

For a symmetric tree, as studied by Gal ([10]), we have \( F_n = R_n \) for all arcs \( a \), and so \( \tau_a \) is simply twice the length of the branch \( \sigma (a) \). For a terminal arc \( a \)
we clearly have $v_a = \tau_a/2$. Then by induction on node levels we have by (3), for any node $y$, that

$$v_y = \frac{\tau_a v_a + \tau_b v_b + \tau_a \tau_b}{\tau_a + \tau_b} = \frac{\tau_a^2/2 + \tau_b^2/2 + \tau_a \tau_b}{\tau_a + \tau_b} = \frac{\tau_y}{2}.$$

So the Value is $V = v_O = \frac{\tau_a^2 + \tau_b^2}{2}$. For binary trees, the probability of taking a branch $a$ is given by $r_a = \frac{\tau_a + \tau_a/2 - \tau_b}{\tau_a + \tau_b} = \frac{\tau_a + \tau_a/2 - \tau_b}{\tau_a + \tau_b} = \frac{1}{2}$. Thus we have Gal’s result,

**Corollary 7** The EBD distribution is optimal for the Hider in the search game on a tree. The Value is the length of the tree.

### 3.4 Weighted Trees

It has been noted that for the weighted trees (with search costs $c_i$, as discussed in the introduction) the optimal hiding strategy (but not the value) is the same as that for the tree with additional arcs of length $c_i/2$ at nodes $i$. Our analysis gives a simple explanation of this fact. We have explained in the Introduction why the game on the weighted tree is equivalent to the game on the tree with additional arcs $a_i$ at $i$ of forward and reverse travel times $F_{ai} = c_i$ and $R_{ai} = 0$, with travel times $\tau_{a_i} = c_i$. But we have also shown that the optimal (EBD) Hider distribution only depends on the travel times. So if we change the times on the additional arcs $a_i$ to $F_{ai} = c_i/2$ and $R_{ai} = c_i/2$, the travel time $\tau_{a_i}$ is still $c_i$, so the EBD distribution for this network is the same.

### 4 Extensions of the Model

In this section we extend our asymmetric formulation to networks other than trees and (in the final subsection), to trees where the Searcher can start anywhere and to mobile Hiders.

We shall not attempt in this paper to give a general formulation of search games with asymmetric arc travel times on all networks. However, in this section we take a look at what happens on the simplest network that is not a tree: a network with a single loop at the start node $O$. In the second part we add an additional ray with symmetric travel time 1, obtaining the lollipop network drawn in Figure 3.

![Figure 3. Lollipop network](image)

We find a new property of the optimal search strategy: the circle has to be searched differently, depending on whether it is searched before or after the line is searched. This differs from the property found earlier for asymmetric trees,
that there was an optimal branching strategy in which subnetworks (subtrees in that case) were searched according to a fixed (branching) behavioral strategy that did not depend on what was already searched.

4.1 The Asymmetric Circle

Consider first just the unit circumference circle network, with clockwise speed 1/2 and anti-clockwise speed 1, and corresponding clockwise and anticlockwise travel times 2 and 1. Parameterize its points as \([0, 1) \mod 1\), with \(x\) denoting the clockwise distance from the root \(O = 0\). For trees, optimal Searcher paths always traversed arcs from one end to the other and the Hider was never in the interior of an arc, so it was sufficient to specify asymmetric distances (travel times) only between nodes. However for general networks we need to specify on all arcs forward and reverse travel times \(F(x)\) and \(R(x)\) to a point at distance \(x\) from its backward and forward ends (arbitrarily allocated), but we start here with the linear case where speeds in each direction are given, so \(F\) and \(R\) are linear in \(x\).

The value \(V\) of the circle network is \(2 = \sqrt{3}\). An optimal strategy for the Searcher is to go all around the circle clockwise (strategy \(C\)) and anticlockwise (strategy \(A\)) with respective probabilities \(1/3\) and \(2/3\); an optimal strategy for the Hider is to hide a clockwise distance \(x\) from the root with probability density \(\check{h}(x) = \frac{2}{3} \cdot \frac{2}{x}\); and hence cumulative distribution \(\check{H}(x) = x \cdot \left(\frac{2}{x} \cdot \frac{2}{x}\right)\).

The stated Searcher strategy finds any point \(x\) in expected time \(1 = 3\) \((2 + \frac{1}{3})\) + \(\frac{1}{3}\) \((1 + \frac{1}{3})\) = \(2\). Against the Hider distribution \(\check{h}\), first consider any contractible search path \(S\) which covers the circle. For any such path there is a unique \(z\) such that all \(x\) in \((0, z)\) (that is, anticlockwise of \(z\)) are reached for the first time from the clockwise direction and all points \(x\) in \((z, 1)\) from the anticlockwise direction. Denote these searches by \(S_c\) (O clockwise to \(z\) and back) and \(S_a\). Their search densities are given by

\[
\rho(S_c(z)) = \frac{\check{H}(z)}{3z} = \frac{2 - z}{3} > \rho(S_a(z)) = \frac{(1 - \check{H}(z))}{3(1 - z)} = \frac{1 - z}{3}.
\]

So by the Search Density Lemma we should carry out \(S_c\) before \(S_a\). Call this strategy \(S(z)\). Hence \(S(z)\) finds points \(x\) in \((0, z)\) at time \(2\), and points \(x\) in \((z, 1)\) at time \(3z + (1 - x)\). So the expected capture time is given by

\[
T(S(z), \check{h}) = \int_0^z 2x \cdot \check{h}(x) \, dx + \int_z^1 (3z + (1 - x)) \cdot \check{h}(x) \, dx
\]

\[
= \frac{2}{3} + z(z - 1)^2, \text{ with}
\]

\[
\inf_{z \in (0,1)} T(S(z), \check{h}) = \frac{2}{3}.
\]

Here the minimum on \([0, 1]\) is attained at \(z = 0\) (corresponding to strategy \(A\)) and \(z = 1\) (corresponding to \(C\)).

4.2 Lollipop Network

Suppose the Searcher adopts the mixed strategy which uses the pure strategies \(LC, LA\) and \(AL\) with probabilities \(1/3, 1/9\) and \(5/9\), respectively. Here \(L\) stands
for the search of the line and $C$ and $A$ stand for the clockwise and anticlockwise traversals of the circle. If the Hider either locates at the end of the line, or at a distance $x$ clockwise of $O$, the expected capture times are both seen to be $14/9$. As the other hiding locations (i.e. on the line but not at its end) are dominated, this shows that $V \leq 14/9$.

For the reverse estimate, suppose the Hider hides at the end of the line with probability $2/3$, and on the circle with probability $1/3$. Given that he is on the circle, he adopts the probability density $\hat{h}$ defined above. Recall that when using this distribution, a Hider on the circle is found by a Searcher starting at $O$ in expected time at least $2/3$. So if the Searcher searches the line first, then the expected capture time is at least

$$\frac{2}{3} \left(1 + \frac{1}{3} \left(2 + \frac{2}{3}\right)\right) = \frac{14}{9}$$

If he searches the circle first, in anticlockwise direction, the expected capture time is at least

$$\frac{2}{3} \left(1 + 1 + \frac{1}{3} \left(\frac{2}{3}\right)\right) = \frac{14}{9}.$$  

If he searches the circle first, but in a clockwise direction, the first `1' becomes a `2' (the new time to traverse the circle), so the time is strictly higher. Thus $V = 14/9$.

It is useful to observe that the hiding probabilities $2/3$ and $1/3$ for the line and circle are proportional to their (minimum) search times of $2$ and $1$. This would seem another example, not on a tree, where the EBD distribution is optimal for the Hider.

### 4.3 Insufficiency of Markovian lollipop strategies

We saw in the previous section that for trees there are optimal Searcher strategies which are branching strategies. This means in particular that once a sub-network (subtree) is reached, it can be searched in a manner independently of when it is reached or of what has been searched before. We now show that this is no longer true for the lollipop network, with respect to the circle subnetwork.

For lollipop, suppose the Searcher is constrained to search the circle in a clockwise manner with some probability $q$, whether it is searched before or after the line. In Section 4.1 we showed for the asymmetric circle network that the $q$ that minimizes the expected time $v_c(q)$ (to find the Hider if he is there) is $q = 1/3$, with $v_c(1/3) = 2/3$, and the corresponding expected search time $\tau$ is $\tau_c(1/3) = 4/3$. The $q$ that minimizes the search time is $q = 0$, with $\tau_c(0) = 1$, and $v_c(0) = 1$. More generally, we have for $0 \leq q \leq 1$ that the expected search time $\tau_c(q)$ and the expected worst case capture time $v_c(q)$ are given by

$$\tau_c(q) = 1 + q \quad \text{and} \quad v_c(q) = \max (1 - q, 2q).$$  

$[v_c(q), \tau_c(q)]$ For the line, we have $v_l = 1$ and $\tau_l = 2$. So if the Searcher must adopt $q$ for searching the circle, independently of when it is searched, the value $V$ of the resulting game on the lollipop is that of the tree shown in Figure 4, where the labels are as shown, and hence $V$ is given by (3) as

$$V(q) = \frac{1 \cdot 2 + \max (1 - q, 2q) \cdot (1 + q) + 2 (1 + q)}{2 + (1 + q)},$$  

drawn in Figure 5.
For $q \leq 1/3$, $V(q) = (-q^2 + 2q + 5) / (q + 3)$, with minimum of 5/3 at $q = 0$ and $q = 1/3$. Since $14/9 < 5/3$, it is clear that the Searcher can only play optimally on the lollipop by using different methods of searching the circle, depending on whether it is searched before or after the line. This is a qualitatively new property.

4.4 Mobile Hider on the Asymmetric Circle

Consider again the asymmetric unit circle network described in the first subsection, except now suppose that the Hider can move continuously. We will not put limits on his maximum speed. Assume that the Searcher can move at speed 1 in the clockwise direction and at speed $\gamma > 1$ in the anticlockwise direction. We parameterize points on the circle by their distance $x$ in $[0, 1)$ from root $O$ in the clockwise direction, which is the same as the time required to reach $x$ in that direction. The case $\gamma = 1$ was proposed by Isaacs [13] and solved by Zeliken [20] and others. Our solution here follows the same idea, so we give only a brief description.

Let $\beta = 1 / (\gamma + 1)$ and let $z_k = k \beta \mod 1$ be points on the circle such that $z_{k+1}$ is distance $\beta$ clockwise of $z_k$, for $k = 0, \ldots, \infty$. Note that it takes the Searcher exactly time $\beta$ to go from $z_k$ to $z_{k+1}$, in either direction. Consider the Searcher mixed strategy $\hat{s}$ which produces paths $S$ with $S(k\beta) = z_k$ and which goes from $z_k$ to $z_{k+1}$ equiprobably in either direction (at maximum speed). Let $T(\hat{s})$ denote the expected time, in the worst case for the Searcher, to reach the Hider when adopting $\hat{s}$. Since either the clockwise or anticlockwise path will intercept the Hider by time $\beta$, and they are chosen equiprobably, we have

$$T(\hat{s}) \leq \frac{1}{2} \beta + \frac{1}{2} (\beta + T(\hat{s})), \text{ or } T(\hat{s}) \leq 2\beta.$$ 

Next, pick a small positive number $\varepsilon$, and let $t_k = k \beta - \varepsilon$. Let $\hat{h}$ denote the Hider strategy which waits at $z_1$ until time $t_1$, and in every time interval $[t_k, t_{k+1}]$ moves from $z_k$ to $z_{k+1}$ equiprobably in either direction, and at the Searcher’s maximum speed in the chosen direction. Observe that if $T > t_k$ only one of the two Hider paths can be intercepted before time $t_{k+1}$, so we have that the expected time $T(\hat{h})$ to find a Hider adopting $\hat{h}$ satisfies

$$\Pr(T \geq t_{k+1} \mid T > t_k) \geq 1/2.$$
Since \( \Pr(T > t_1) = 1 \), we have \( \Pr(T > tk) \geq 1/2^k \), and so the expected value of the capture time \( T \), given that the Hider adopts \( \hat{h} \), is at least (in the limit where \( \varepsilon \to 0 \))

\[
\beta \sum_{i=1}^{\infty} \Pr(T \geq i\beta) = \beta \sum_{k=0}^{\infty} 2^{-k} = 2\beta, \quad \text{which is the Value.}
\]

In the symmetric case \( \gamma = 1 \) we have \( \beta = 1/2 \), so the value is 1.

The point of this exercise is to show that, unlike the case for an immobile Hider, the theory does not qualitatively change when for mobile Hiders when asymmetry of travel times is introduced. (In particular, the equiprobable use of the two search directions still holds.)

### 5 Arbitrary Searcher Starting Point

The search game for an immobile Hider can also be studied under the assumption that the Searcher can pick his starting point, in which case the search space \( Q \) is simply a network, rather than a rooted network. We assume the Hider can pick any leaf node as \( H \), so the Searcher will also start at a leaf node. We will adopt the notation \( \text{EBD}(y) \) for the EBD distribution on the rooted network \( Q, y \), where \( y \) is a point in \( Q \).

Let \( \tau \) denote the minimum time to tour \( Q \), the sum of \( F_a + R_a \) over all its arcs \( a \).

For (symmetric) trees, Dagan and Gal [9] showed that

- an optimal strategy for the Hider is \( \text{EBD}(c) \), where \( c \) is the center, the point which minimizes the maximum distance (travel time) between \( c \) and other points in \( Q \).
- the Searcher can restrict to paths which start and end at two diametrical points (at maximum distance). (In fact he can mix equiprobably between such a path and its reverse path.)

Even for asymmetric stars (trees of level 1), neither of their results is always true, although analogs to the asymmetric case may sometimes hold. Letting \( d(x,y) \) represent the asymmetric distance given by the travel time from \( x \) to \( y \), we define the \( \text{out-center} \) \( c^o \) to be a point which minimizes the maximum distance to other points, that is,

\[
\max_y d(c^o, y) = \min_x \max_y d(x, y) \equiv \text{radius}(Q). \tag{6}
\]

We now assume that \( Q \) is a star, with a central node \( O \) and leaves \( i, i = 1, \ldots, n \), which have forward and reverse (with respect to \( O \)) travel times \( F_i \) and \( B_i \), with the \( F_i \) decreasing. Denote the arc between \( O \) and \( i \) by \( a_i \), assume these are parameterized by a number \( x \) which is 0 at \( O \) and 1 at \( i \). Thus \( c^o \) lies on arc \( a_1 \) for an \( x \) satisfying

\[
(1 - x) F_1 = x R_1 + F_2, \quad \text{or} \quad x = \frac{F_1 - F_2}{F_1 + R_1}. \tag{7}
\]

(The left side of the equation represents the travel time \( d(x,1) \) to node 1 and the right side to node 2, the latter involves travel on arc \( a_1 \) in reverse and then
on arc $a_2$ forwards.) If $n = 2$ it is easily seen that $EBD(c^0)$ is optimal for the Hider and

$$ V = (F_1 + R_2)(F_2 + R_1) / (F_1 + F_2 + R_1 + R_2). \tag{8} $$

However for $n > 2$, $EBD(c^0)$ may or may not be optimal for the Hider, and the Searcher may or may not be able to restrict to paths between nodes at distance of radius $Q$ from $c^0$. To see this, consider the stars drawn in Figure 6, where the optimal hiding distributions (calculated from LP’s) are indicated next to the node number. For the moment, ignore the indicator $I$ which will be defined later to distinguish between cases.

![Figure 6. Three asymmetric star networks, $I > 0$, $I < 0$, $I = 0$.](image)

In the star on the left, the optimal Hider strategy is given by $EBD(c)$, where $c$ is the out-center, located 1/3 the way out along arc $a_1$. For example, the time to tour from $c$ to 1 and back is $(2/3)(7 + 5) = 8$, compared with a time of 20 to tour the network. Note that the probabilities $9/20$ and $3/20$ on the other arcs are in proportion to their travel times (6 and 2) from the non-leaf node. However the Hider strategy for the middle star cannot be $EBD(y)$ for any point $y$ because no two nodes have probabilities proportional to the travel times of their arcs, and if $y$ is on some arc, the distribution on the other arcs must be EBD (suitably normalized). Even in the left-hand case, where there is an EBD distribution which is optimal for the Hider, the Dagan-Gal result that the Searcher can always employ paths which start and end at a pair of diametrical points (here 1 and 2) no longer holds. The optimal Searcher strategy searches the nodes in orders 132, 231, and 123 with respective probabilities $10/20$, $9/20$ and $1/20$, and the value is $27/5 = 5.4$. The best $T$ he can guarantee is with $p = 5/11$, with $T = 60/11 \approx 5.45$ which is longer than Value of $27/5 = 5.4$. The fact that the Searcher needs to use only strategies 132, 231 and 123 is easily explained in terms of the Search Density Lemma, as shown below.

However in the star on the right, where EBD is also optimal, it is easily seen that the two diametrical paths 132 and 231 reach nodes 1, 2, 3 in respective times $(0, 10, 6)$ and $(10, 0, 4)$, so their equiprobable average reaches them in times $(5, 5, 5)$, where $V = 5$. It turns out that this property is non generic for asymmetric stars, that is, it holds only on a lower dimensional space of the parameters. Arbitrarily small perturbations of any travel time of an arc, either increasing or decreasing it, lead to stars that cannot be optimally searched using only diametrical paths.
To deal with general stars where \( n > 3 \), we amalgamate the arcs \( a_i, i > 2 \), into a star we call \( Q_3 \). We denote the value of this star (with Searcher start at root \( O \)) as \( v_3 \), and its search time by \( \tau_3 \). (When \( n = 3 \) we have \( v_3 = F_3 \) and \( \tau_3 = F_3 + R_3 \).) The following lemma characterizes optimal response paths to the EBD Hider strategy \( h \).

**Lemma 8** Let \( Q \) be a star network with non-leaf node \( O \) and leaf nodes \( i = 1, 2, \ldots, n \) labelled so that the \( F_i \) are decreasing. Let \( h = \text{EBD}(c^o) \), where \( c^o \) is the out-center. Any search path which is an optimal response to \( h \) either (i) starts at 1, or (ii) starts at 2 and ends at 1. Such paths have expected capture time \( \bar{V} \), where

\[
\bar{V} = \frac{(\tau_3 + F_1 + R_2) (v_3 \tau_3 + F_2^2 + \tau_3 F_2 + \tau_3 R_1 + \tau_3 R_2 + F_2 R_1 + F_2 R_2 + R_4 R_2)}{(\tau_3 + F_2 + R_2) \tau}.
\]

So \( V \geq \bar{V} \), with equality iff \( h \) is optimal.

**Proof.** Suppose \( S \) is optimal against \( h \), as it must be to have positive probability in an optimal mixed strategy. Let \( S_i \) denote a path starting at node \( i \). Suppose \( i \neq 1 \). Then once \( S_i \) reaches \( O \), the tours of the arcs \( a_j, j \neq 1 \) or \( i \), all have the same search density, which is greater than the search density of \( a_1 \). So by the Search Density Lemma, \( S_i \) must search node 1 last. For \( i, j \neq 1 \), we have for searches \( S_i \) starting at \( i \) and ending at 1, that

\[
T(S_i, h) - T(S_j, h) = F_j - F_i,
\]

so the optimal one in this class is \( S_2 \), as claimed. The formula for \( \bar{V} \) is easily calculated from the matrix multiplication \( (h(1), h(2), h(3)) M = (\bar{V}, \bar{V}, \bar{V}) \), where \( M \) is given below in (10).

This leads to the questions of when \( h \) is optimal and when the Searcher can concentrate on Chinese postman paths (not taken equiprobably though). We can answer both questions in terms of an indicator function \( I \) defined by

\[
I \equiv F_1 F_2 - R_1 R_2 + \tau_3 (F_1 + F_2 + \tau_3) - v_3 (F_1 + F_2 + R_1 + R_2 + 2\tau_3).
\]  

**Theorem 9** Let \( Q \) be a star with \( n \geq 3 \) arcs with \( F_i \) decreasing. Then

1. If \( I > 0 \) then the EBD Hider strategy is optimal, and the Searcher has no optimal strategy concentrated on paths starting and ending at the diametrical nodes 1 and 2.

2. If \( I = 0 \) then the EBD Hider strategy is optimal, and the Searcher has an optimal strategy which always starts and ends at the nodes 1 and 2.

3. If \( I < 0 \) then the EBD Hider strategy is not optimal.

**Proof.** If EBD is optimal then the previous result shows that the Searcher has an optimal strategy which randomizes over the following three mixed strategies \( s_1, s_1^* \) and \( s_2 \): \( s_1 \) starts at 1, searches \( Q_3 \) from \( O \) in an optimal mixed strategy, then goes to 2; \( s_1^* \) is the same but searches \( a_2 \) before \( Q_3 \); \( s_2 \) starts at 2, then searches \( Q_3 \) optimally and searches 1 last. The expected capture times for these
strategies, for a hider at 1, 2, or one hiding optimally in $Q_3$, $O$ are given by the matrix

\[
\begin{array}{ccc}
1 & s_1 & 0 \\
2 & R_1 + \tau_3 + F_2 & 0 \\
Q_3 & R_1 + v_3 & R_2 + v_3 \\
\end{array} = \begin{array}{c}
s_2 \\
R_2 + \tau_3 + F_1 \\
R_1 + F_2 \\
\end{array}
\] (10)

If we seek probabilities for the columns which equalize payoff for the rows (at $V$), we find that without the non-negativity condition of the probabilities, we get (with $k = \tau (\tau - F_1 - F_2)$)

\[
\begin{align*}
\Pr (s_1) &= \frac{\tau (R_2 + v_3)}{k}, \\
\Pr (s_2) &= \frac{F_2 (F_2 + R_1 + R_2 + \tau_3) + R_1 R_2 + \tau_3 (R_1 + R_2 + v_3)}{k}, \\
\Pr (s_1^*) &= \frac{I}{k}.
\end{align*}
\] (11)

The first two numbers are always non-negative. The last, $\Pr (s_1^*)$, will be a probability if and only if $I \geq 0$. So if $I < 0$, we have a contradiction to the Lemma, establishing part 3.

If $I \geq 0$, then the mixed strategy given by solution shows that $V \geq \bar{V}$ and hence by the Lemma that $V = \bar{V}$ and EBD is optimal for the Hider, establishing part 1. Finally, if $I = 0$, part 2 holds (since also $I \geq 0$) and since $\Pr (s_1^*) = 0$ the optimal Searcher strategy of (11) uses only $s_1$ and $s_2$, both of which start and end at the diametrical nodes 1 and 2. This establishes part 3.

For the left and middle stars of Figure 6, we have $v_3 = F_3 = 1$, $\tau_3 = F_3 + R_3 = 2$ and $\tau = 20$, so the indicator function $I$ reduces to $4F_1 - 4R_1$, which is $I = 5 \cdot 7 - 3 \cdot 5 - 12 = 8$ for the left hand star, and $I = 5 \cdot 5 - 3 \cdot 7 - 12 = -8$ for the middle star. More generally, we note that the indicator function is increasing in $F_i - R_i$ for $i = 1, 2$, and decreasing in $2v_3 - \tau_3$ (which is $F_3 - R_3$ for the $n = 3$ arc case). So for example the middle star of Figure 6 can be obtained from the symmetric tree with $F_1 = R_1 = 6$ by decreasing $F_1$ and increasing $R_1$, and thus decreasing $F_1 - R_1$. Hence the asymmetric tree does not have an EBD strategy which is optimal. The star on the right has $I = 0$, and we have already shown how an equiprobable mixture of search paths 132 and 231 guarantees the Value of 5. For symmetric stars, we obtain the following result of Dagan and Gal.

**Corollary 10** For symmetric stars, EBD($c$) is optimal for the Hider and the Searcher can concentrate on paths between diametrical points.

**Proof.** For symmetric stars Gal’s result for symmetric rooted trees says that $\tau_3 = 2v_3$, so

\[
I = F_1 F_2 - R_1 R_2 + v_3 ((F_1 - R_1) + (F_2 - R_2)) = 0,
\] (12)

and the result follows from part 2 of the Theorem.

As in Dagan and Gal [9], we can compare the values of the arbitrary and fixed start games, in terms of the radius (now the out-radius), as

\[
V (Q) \geq V (Q, c^0) - \text{radius} (Q).
\]
To see this, suppose the Searcher knows an optimal mixed strategy for $V(Q)$. If he has to start at $c^0$ he can follow this strategy, delayed by time $\text{radius}(Q)$, by arriving at the various starting points at that time and then following that strategy. So $V(Q, c^0)$ is no more than $V(Q) + \text{radius}(Q)$.

For any star $Q$, let $\Gamma^*$ be the game where the Hider can hide at $O$ or at any leaf node, and let $V^*$ be the Value of this game. For example, for a star with two arcs from non-leaf node $O$, with $F_1 = 5$, $R_1 = 7$, $F_2 = R_2 = 3$, the optimal Hider strategy $h$ for $\Gamma^*$ is $h(1) = h(O) = 5/12$ and $h(2) = 1/6$, with $V^* = 55/12 \approx 4.58$ instead of $V = (5 + 3)(7 + 3)/18 = 459/18 \approx 4.44$ as given by (8). The Searcher employs $102, 201$ and $O12$. We can explain why the Hider would not need to hide at non-leaf nodes in the symmetric case by recalling in that case $I = 0$ and applying the following result.

**Theorem 11** if $I(Q) \geq 0$, then $V^*(Q) = V(Q)$, so the Hider does not need to hide at the non-leaf node $O$.

**Proof.** First note that

$$V(Q) \leq V^*(Q) \leq V(Q_\varepsilon),$$

(13)

where $Q_\varepsilon$ is the star obtained from $Q$ by adding an additional leaf node $n + 1$, with $F_{n+1} = R_{n+1} = \varepsilon$. The left inequality follows from the fact that the Hider has an additional pure strategy (hiding at $O$) in the game $\Gamma^*$. The right inequality follows from the observation that the Value is determined, in both cases, from the $n + 1 \times n + 1$ matrix of asymmetric distances $d(i, j)$ where $i, j$ represent the hiding nodes. Identifying node $O$ of $Q$ with node $n + 1$ of $Q_\varepsilon$, we see that the distances are the same unless $i$ or $j \neq i$ are $O$ (or $n + 1$), in which case the distances for $Q_\varepsilon$ are $\varepsilon$ larger. (In other words a hiding strategy on $Q$ does better on $Q_\varepsilon$ if the probability of hiding at $O$ is moved to node $n + 1$.)

If $I(Q) \geq 0$, then $h(Q_\varepsilon, c^\varepsilon) \equiv h_\varepsilon$ is optimal on $Q_\varepsilon$ by Theorem 9, where $c^\varepsilon$ is independent of $\varepsilon$. So $h_\varepsilon(n + 1) \leq 2\varepsilon/(\tau_3 + 2\varepsilon)$, by the definition of EBD, which goes to zero. Hence $V(Q_\varepsilon) \rightarrow V(Q)$, since the Searcher can play optimally in $Q$ and then go to node $n + 1$. So (13) gives $V(Q) = V^*(Q)$. ■

### 6 Conclusions

This paper extends the theory of network search games to networks with asymmetric travel times, mainly trees. For trees, some parts of the theory established by Gal [10] still hold, but other parts fail. The EBD distribution, suitably interpreted, remains the unique optimal distribution for the Hider. The independence of value on the starting (root) location is no longer true, and there are no longer necessarily optimal Searcher strategies consisting equiprobably of a tour and its reverse. Instead, this paper highlights the importance of branching strategies for trees. However for the lollipop network, there are no longer optimal search strategies which (like branching strategies) search subnetworks in the same stochastic manner, independently of when they are searched. No really new phenomena were found when introducing asymmetric travel times in the search for a mobile Hider on the circle, merely small modifications of the optimal strategies for the symmetric case.

From the computational viewpoint, this paper gives a simple recursive method for calculating the value and optimal strategies for the search game on trees with
asymmetric travel times. We also show how to incorporate search costs at nodes into (asymmetric) travel times on an additional arc.

An important question raised by this analysis is the extent to which our techniques can be extended to networks other than trees; a first example in this quest was analyzed in Section 4, but the general question remains wide open. Similarly, the analysis of trees with arbitrary starting points has only been briefly touched upon here, and is currently under study.

References


