Steve Alpern, Thomas Lidbetter

Mining coal or finding terrorists: the expanding search paradigm

Article (Accepted version)
(Refereed)

Original citation:
Alpern, Steve and Lidbetter, Thomas (2013) *Mining coal or finding terrorists: the expanding search paradigm.* Operations Research, 61 (2). pp. 265-279. ISSN 0030-364X

DOI: 10.1287/opre.1120.1134

Abstract

We show how to optimize the search for a hidden object, terrorist, or simply Hider, located at a point H according to a known or unknown distribution ν on a rooted network Q. We modify the traditional 'pathwise search' approach to a more general notion of 'expanding search'. When the Hider is restricted to the nodes of Q, an expanding search S consists of an ordering $(a_1, a_2, \dots)$ of the arcs of a spanning subtree such that the root node is in a_1 and every arc a_i is adjacent to a previous arc $a_j,\ j < i$. If a_k contains H, the search time T is $\lambda(a_1) + \cdots + \lambda(a_k)$, where λ is length measure on Q. For more general distributions ν, an expanding search S is described by the nested family of connected sets $S(t)$ which specify the area of Q that has been covered by time t. $S(0)$ is the root, $\lambda(S(t)) = t$, and $T = \min\{t : H \in S(t)\}$. For a known Hider distribution ν on a tree Q, the expected time minimizing strategy $\bar{S}$ begins with the rooted subtree Q' maximizing the 'density' $\nu(Q')/\lambda(Q')$. (For arbitrary networks, we use this criterion on all spanning subtrees.) The search $\bar{S}$ can be interpreted as the optimal method of mining known coal seams, when the time to move miners or machines is negligible compared to digging time. When the Hider distribution is unknown, we consider the zero-sum search game where the Hider picks H, the Searcher S and the payoff is T. For trees Q, the value is $V = (\lambda(Q) + D)/2$, where D is a mean distance from root to leaf nodes. If Q is $2-$arc connected, $V = \lambda(Q)/2$. Applications and interpretations of the expanding search paradigm are given, particularly to multiple agent search.

Mining Coal or Finding Terrorists: The Expanding Search Paradigm

Steve Alpern and Tom Lidbetter
London School of Economics

March 2, 2012

1 Introduction

The usual search paradigm when seeking a stationary target on a network is what we now call *pathwise search*, where the Searcher follows a continuous unit speed path until the target is reached. This article introduces a new search paradigm, which we call *expanding search*, where the Searcher may restart the search at any time from any previously reached point. Such searches are routinely carried out in many contexts, sometimes by a team of agents. Under such searches, the portion $S(t)$ of the search region that has been covered by time t expands in a continuous way until the first time T (the search time) that it contains the target of the search. This is the first article to give a rigorous definition and analysis of expanding search on a network that is comparable to the well known theory developed for pathwise search by Gal (1979, 2000, 2011), Baston and Bostock (1990), Anderson and Aramendia (1990), Kikuta and Ruckle (1994), Pavlovic (1995), Reynierse and Potter (1993), Alpern and Gal (2003), Jotshi and Batta (2008), Alpern (2010, 2011 a,b). The work of Alpern and Howard (2000) considered a related problem of a single Searcher who alternates between looking for a single Hider at two locations, which can now be seen as a special case (on a star network) of expanding search. There is also a connection to the two-Searcher *coordinated search* problem of Thomas (1992) and Reyniers (1995,1996) described in the next section. Megiddo et al (1988) considered minimizing the number of searchers, rather than the search time. Other interpretations of expanding search, such as the optimal mining of coal, will be described later.

An immobile hidden object, target, terrorist, or simply Hider, is located at an unknown point H of a known network Q. The network is endowed with an arc length measure λ (linear Lebesgue measure) so that $\lambda(a)$ denotes the length of an arc a and $d(x, y)$ is the metric given by the length of the shortest path between points x and y in Q. We assume there are a finite number of arcs, each of finite length, so that Q is compact. The distribution (probability measure on Q) ν of H may be known or unknown. If it is known, we consider the *Bayesian Search Problem* of minimizing the expected search, or capture, time. If it is unknown, we consider the zero-sum *Expanding Search Game* where the Hider chooses H to maximize T. Starting at a given point of Q, called the root, a search team consisting of successively dividing groups spreads out over the network until the first (capture) time T that one of its members encounters the Hider. The agents are constrained to move with combined speeds of 1. This means that $\lambda(S(t)) = t$, where $\lambda(S(t))$ is the measure of the portion of Q covered by time t. When the Hider distribution ν is given on any network, we solve the Bayesian Problem by an algorithm and a heuristic that determine the expanding search $S(t)$ that minimizes the expected value of T. We solve the Search Game when Q is a tree and give bounds on the value for arbitrary networks Q.

In the important case where the Hider distribution ν is concentrated on the nodes of Q, an expanding search is simply a sequence of arcs $S = (a_1, a_2, \ldots, a_N)$, oriented so that the tail of a_1 is the root O of Q, the tail of every other arc a_i is the tip of a previous arc a_j, $j < i$, and the N non-root nodes of Q coincide with the N tips of the arcs. If the Hider location H is the tip of arc a_k, the capture time $T = T(S, H)$ is given by

$$T(S, H) = \lambda(a_1) + \cdots + \lambda(a_k). \tag{1}$$

A simple but important concept used in the solution of both the Bayesian and Search Game Problems is that of the *search density,* or simply *density,*

$$\rho(K) = \nu(K) / \lambda(K) \tag{2}$$

of a connected subset K of Q: the probability that H will be found in K divided by the time required to search K (which for expanding search is simply its total length $\lambda(K)$). We show (Theorem 3) that when Q is a rooted tree and K is a subtree of maximum search density for some known Hider distribution ν, there is an optimal expanding search $\bar{S}(t)$ that begins by exhaustively searching K, that is, with $\bar{S}(t) = K$ at time $t = \lambda(K)$. While this optimality condition does not also hold for arbitrary networks, we can however use it to solve the Bayesian Search Problem on any network by considering its spanning trees.

We are able to give a complete solution of the Expanding Search Game for any rooted tree Q. We show (Theorem 18) that value of this game is given by

$$V = \frac{\mu + D(Q)}{2},$$

where $\mu = \lambda(Q)$ is the total length of the tree Q and $D(Q)$ is the mean distance from the root node O of Q to its leaf nodes, with respect to the Equal Branch Density (EBD) distribution e on the leaf nodes. This is the unique distribution such that at every branch node of the tree, the search density of each branch is the same. We determine the optimal Searcher mixed strategy as a branching function which specifies the probability that each branch at a node should be searched first when reaching that node, regardless of how the search has proceeded up to that point.

If the network is 2-arc connected (removing an open arc never disconnects it) then we show that the value of the expanding search game is $\mu/2$. In this case the uniform distribution is optimal for the Hider and an equiprobable mixture of a certain search and its reverse time search is optimal for the Searcher. We also derive bounds on the value V for arbitrary networks. Some of these apply as well to the pathwise version.

To illustrate these ideas, consider the tree Q with root O depicted in Figure 1. We will use this network as an example at several points throughout this paper, not always assigning the same lengths to the arcs.

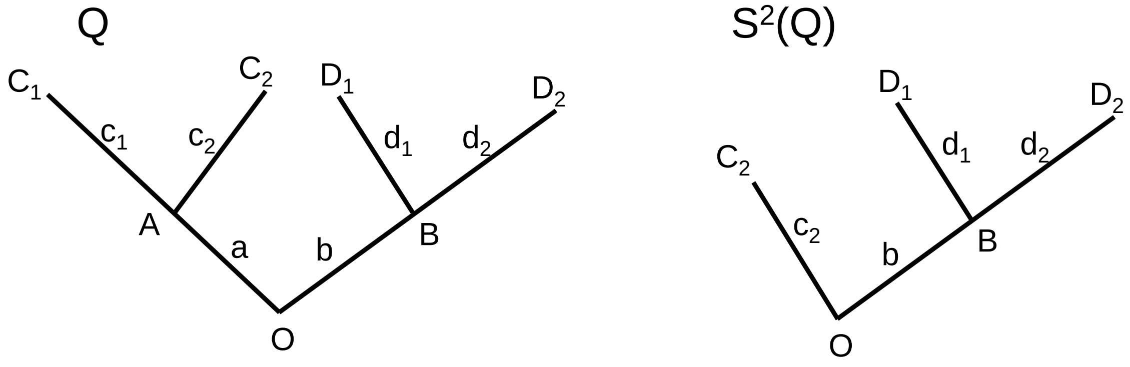

Figure 1. A rooted tree Q and its contraction $S^2(Q)$.

An example of an expanding search on Q is $S = (b, d_1, a, d_2, c_2, c_1)$. Consider the network Q_0 given by the following choice of arc lengths that is symmetric in the c_i and the d_i:

$$\lambda(a) = 1, \lambda(b) = 2, \lambda(c_1) = \lambda(c_2) = 1, \lambda(d_1) = \lambda(d_2) = 5 \tag{3}$$

For the search S, the time taken to reach, say C_2 is given by $T(S, C_2) = 2+5+1+5+1 = 14$, using the notation T for the search time introduced in (1). Note that the search S arrives at nodes A, B, C_1, C_2, D_1 and D_2 in respective times $8, 2, 15, 7, 13$ and 14. Hence if, for example, the Hider distribution ν takes the value $\frac{2}{7}$ on C_1 and $\frac{1}{7}$ on each of the other non-root nodes, then the expected search time is $T(S, \nu) = \frac{1}{7} \cdot 8 + \frac{1}{7} \cdot 2 + \frac{2}{7} \cdot 15 + \frac{1}{7} \cdot 7 + \frac{1}{7} \cdot 13 + \frac{1}{7} \cdot 14 = 7\frac{4}{7}$, using the notation $T(S, \nu)$ formally defined later in (7). In the case of this Hider distribution, there is a unique rooted subtree M of maximum density, $M = ac_1$, which has density $(\frac{1}{7} + \frac{2}{7})/2 = \frac{3}{14}$. As we will show later in Theorem 3, this indicates that any optimal search must begin by searching M. After searching a and c_1 it is clear that an optimal search must continue by optimally searching the tree depicted on the right of Figure 1. This tree, which is obtained by contracting the first two arcs of Q traversed by this particular search S, shall be referred to as $S^2(Q)$ in Section 3. We can therefore apply Theorem 3 again by seeking the rooted subtree of maximum density in this new network. This is simply c_2, which has density $(\frac{1}{7})/1 = \frac{1}{7}$. The only arc available to search next is b, after which by symmetry d_1 and d_2 can be searched in either order. Hence an optimal search of Q_0 is $S_{opt} = (a, c_1, c_2, b, d_1, d_2)$, which searches the nodes $a, b, c_1, c_2, d_1,$ and d_2 in respective times $1, 5, 2, 3, 10,$ and 15, and hence has expected search time $T(S_{opt}, \nu) = \frac{1}{7} \cdot 1 + \frac{1}{7} \cdot 5 + \frac{2}{7} \cdot 2 + \frac{1}{7} \cdot 3 + \frac{1}{7} \cdot 10 + \frac{1}{7} \cdot 15 = 5\frac{1}{7}$.

Turning now to the *game* played on Q_0, we first note that any optimal Hider strategy must be restricted to the leaf nodes, C_i and D_i of Q_0, since all other points are dominated.

Hence each player has a finite set of undominated pure strategies, and the game can be reduced to a matrix game. We note further that because of the symmetry in the network between the C_i and the D_i, it is clear that in the Hider's optimal mixed strategy he must choose equiprobably between the two nodes in each pair C_i and D_i, and in the Searcher's optimal strategy he must choose equiprobably which order to search the arcs c_1 and c_2 and the arcs d_1 and d_2. Hence we can simplify the matrix game by taking averages: for example in the matrix below, the entry for $(H, S) = (C, (b, d, a, c, c, d))$ corresponds to the average time taken to reach C_1 and C_2 by all four Searcher strategies $(b, d_1, a, c_1, c_2, d_2)$, $(b, d_1, a, c_2, c_1, d_2)$, $(b, d_2, a, c_1, c_2, d_1)$, and $(b, d_2, a, c_2, c_1, d_1)$, that is $\frac{1}{2}((2+5+1+1)+(2+5+1+1+1)) = 9.5$.

	accbdd	*acbdcd*	*acbddc*	*bdaccd*	*bdacdc*	*bddacc*
C	2.5	6	8.5	9.5	12	14.5
D	12.5	12	11.5	11	10.5	9.5

(4)

Solving this matrix game numerically, we find that the Hider's optimal mixed strategy is to choose C with probability $\frac{1}{5}$ and D with probability $\frac{4}{5}$: that is C_1 and C_2 each with probability $\frac{1}{10}$ and D_1 and D_2 each with probability $\frac{2}{5}$. This is an example of the EBD distribution, as formally defined in Definition 5. Notice that the branch $\{a, c_1, c_2\}$ has density $(\frac{1}{10} + \frac{1}{10})/3 = \frac{1}{15}$, and the branch $\{b, d_1, d_2\}$ has equal density $(\frac{2}{5} + \frac{2}{5})/12 = \frac{1}{15}$. The Searcher's optimal strategy is to use the search (a, c, c, b, d, d) with probability $\frac{1}{3}$ and (b, d, d, a, c, c) with probability $\frac{2}{3}$. The game has value $V(Q_0) = 10.5$. As we shall see later in Theorem 18, this value is equal to half the sum of the total measure of the network, $\mu = 1 + 2 + 1 + 1 + 5 + 5 = 15$ and the quantity $D = D(Q_0) = \frac{1}{10} \cdot 2 + \frac{1}{10} \cdot 2 + \frac{2}{5} \cdot 7 + \frac{2}{5} \cdot 7 = 6$, the mean distance of the root node to the leaf nodes with respect to the EBD distribution. That is, $V(Q_0) = \frac{1}{2}(\mu + D) = \frac{1}{2}(15 + 6) = 10.5$.

2 Interpretation and Applications of Expanding Search

The interpretation of an expanding search strategy in terms of a team of pathwise search agents is as follows. In the case where H is a node of Q, assume a group of m search agents starts at the root node. Then a large enough (for later branching) subgroup takes initial arc a_1, while the rest remain at the root. Whenever a new arc a_k is chosen, some Searchers move along it, while some stay at its tail. An interesting problem which involves this group interpretation is to determine the number of agents that are required; either for a particular search strategy or for an optimal one. For trees, clearly the number of leaf nodes is a sufficient number of searchers. For example, the expanding search $(a, c_1, b, d_1, c_2, d_2)$ mentioned in the matrix (4) can be carried out by $m = 4$ agents, each adopting a pathwise search which includes waiting, as indicated by the left table in (5), where for example the fourth search agent follows the pathwise search a, w, w, w, c_2 (where w indicates waiting time). If only $m = 3$ agents are available, at least one of these must go backwards on an arc (indicated by the * in c_1^*) as in the table on the right.

a	c_1	b	d_1	c_2	d
$1, 4$	1	$2, 3$	2	4	3

a	c_1	b	d_1	c_1^*	c_2	d_2
1	1	$2, 3$	2	1	1	3

(5)

If a bound were put on the number of searchers, the expected time would be decreasing in the bound. When the bound is 1 (single agent), we get the usual pathwise search value, where a single Searcher moves at unit speed along the network. For sufficiently large bounds, we get the value obtained here for expanding search. For example, if the network is a line, with the root in the interior, then $m = 2$ searchers are enough for expanding search, whereas a single Searcher is faced with the well known linear search problem. A variation on the two-agent problem on the line, known as *coordinated search*, has been studied by Thomas (1992) and Reyniers (1995,1996), under the interpretation of 'how to find your children when they are lost'. Here the quantity to be minimized is the time when the two agents (parents) are back at the root (their car) after one of them has found the Hider (child). If the Hider is mobile, the question of how many searchers are needed to guarantee finding him is considered in the classic paper of Megiddo et al (1988).

Another interpretation, involving a single 'Searcher' (actually researcher) is that of tackling, say a mathematical problem for which one can map out in advance which facts need to be checked (calculated) before going onto the next step, perhaps seeking a proof or counterexample to a finite conjecture. Some steps of this process must be completed before other steps should be carried out, perhaps first checking the 'obvious' counterexample or proof. We assume here that after one branch of the research tree ends in failure, the researcher can go without time loss back to a previous method of attack. In some contexts the researcher here could be replaced by a computer program, so that we are in fact seeking a program which minimizes the expected time to resolve some question.

The Bayesian problem for a tree, with a known distribution ν, has an interpretation in terms of optimal mining of coal. Suppose that by seismic analysis the density of coal along the tree network of its veins is known. We assume that time or effort involved in moving mining equipment (or miners) along the dug out regions (corresponding to what we called $S(t)$) of the mine is negligible with respect to the digging effort. So expanding search consists of digging in an area for a while, then moving the effort (miners or digging machines) to another area, starting from the last bit dug. The probability density function of the capture time T when ν is the Hider distribution is the same as the extraction rate of coal at time T. So an optimal expanding search strategy corresponds to an extraction policy which minimizes the mean time that the coal is ready for sale. This would make sense in an environment where the discount rate, used in calculating the net present value of coal mined over time, is linear. This is a reasonable approximation in certain economic conditions. Our optimality condition is to mine veins leading to the highest coal density, a semi-greedy policy. If there are enough excavators, it would be possible to always leave one at the last place mined along any vein. Our restriction on the rate of increase of the covered portion might relate to the number of miners or a constraint on the maximum electrical power available.

2.1 Relation to path-planning problems

Our expanding search model shares some elements with problems of *path planning* (e.g. Lavalle, 2006) in that we construct nodes sets, beginning with a starting node s (called O) and always take as our next node one which is adjacent to some previous node. As explained above, this differs from previous Searcher strategies in the field of *search games*, where (in the discrete formulation) each successive node must be adjacent to the previous one, so that repetition of nodes is generally required. However path planning differs from our searcher strategies in certain significant aspects. For comparison, a useful description of path planning problem is given in Dechter and Pearl (1985):

> "Given a weighted directional graph G with a distinguished start node s and a set of goal nodes r, the optimal path problem is to find a least-cost path from s to any member of r where the cost of the path may, in general, be an arbitrary function of the weights assigned to the nodes and branches along that path. A general bestfirst (GBF) strategy will pursue this problem by constructing a tree T of selected paths of G using the elementary operation of node expansion, that is, generating all successors of a given node. Starting with s, GBF will select for expansion that leaf node of T that features the highest 'merit,' and will maintain in T all previously encountered paths that still appear as viable candidates for sprouting an optimal solution path. The search terminates when no such candidate is available for further expansion, in which case the best solution path found so far is issued as a solution; if none has been found, a failure is proclaimed."

Our game theoretic formulation (the main part of the paper) is of course very different from the above, but even our Bayesian model, with a known Hider distribution, is significantly different, in the following ways.

- Our expanding set must reach *all* of the goal nodes r (in our case the support of the Hider distribution, which is not even necessarily a discrete set). The solution to our problem is not a path in G (though it can be interpreted as a walk in G).
- We do not present a local algorithm at all, the next node chosen depends in general on distant effects. We do not even give a global algorithm - our approach is not algorithmic (though this is a possible area for future work).

- Our problem is a stochastic problem, where the value to be minimized is an expected value.

To put the different approaches in the starkest terms, consider the case where there is a *unique* goal node r in the quoted problem and this node is the unique Hider location in our problem. The solution to the path planning problem is a shortest path algorithm for a single goal whereas we would simply say that the Hider should take the shortest path to the Hider node, without providing or considering the process of *finding* that path. Also note that in the above quoted problem, the ‘search’ is for an optimal path while in our problem the ‘search’ is for the unknown node (or point in an arc) containing the Hider. Another main difference is that our Hider may hide in a continuum of locations (any point on any arc), not just at nodes, so our problem is not a finite problem.

3 Known Hider Distribution on Nodes

We begin our analysis of expanding search in the simplest case where the Hider distribution ν is concentrated on the nodes of a network Q. In this case we will consider *expanding arc sequences*, sequences of arcs ordered and oriented so that the tail of each arc is the tip of a previous one. In this section an expanding search S, sometimes referred to as just a *search,* is simply an expanding arc sequence $(a_1, \ldots, a_N)$ where the tail of a_1 is the root node and tips of the arcs are the N non-root nodes of Q. We will refer to a search which is an expanding arc sequence as a *combinatorial search.* If S is an expanding search of Q and $k = 0, \ldots, N$, we denote by $S_k = S_k(Q)$ the rooted subnetwork (it is a tree) of Q formed by the first k arcs searched by S. In particular, every search S of Q is associated with a spanning tree S_N. Note that in the tree S_k, the *ordering* of the arcs in S has been lost. We also let $S^k = S^k(Q)$ be the rooted network formed by shrinking S_k to O in Q. In Section 1 we saw $S^2(Q)$ depicted in Figure 1. After the Searcher has chosen the first k arcs to search, the problem that remains is how to search the remaining network S^k, so using dynamic programming we observe that an optimal expanding search S must be optimal for all of the subproblems (subnetworks) S^k. Given any expanding search $S = (a_1, \ldots, a_N)$ and any Hider location at the tip A_k of arc a_k, the search time $T = T(S, H)$ is given, as in (1), by

$$T(S, H) = \lambda(a_1) + \cdots + \lambda(a_k), \tag{6}$$

and the expected time for S to find a Hider hidden according to distribution (measure) ν is denoted by

$$T(S, \nu) = \sum_{k=1}^{N} \nu(A_k) \cdot T(S, A_k). \tag{7}$$

For a given Hider distribution ν, we say that $\bar{S}$ is optimal (against ν) if it minimizes the expected search time $T(S, \nu)$ over all expanding searches S. (In the context of this section, there are only finitely many expanding searches, so the existence of optimal ones is not in question.)

3.1 Properties of the search density ρ

As mentioned in the Introduction, if we have a fixed Hider distribution (measure) ν on a network Q, we can define the *search density*, or simply *density* $\rho = \rho(A)$ for any subset A of Q by $\rho(A) = \nu(A) / \lambda(A)$. When disjoint regions can be searched in either order, it is well known that it is better to search the region of higher density first. A simple form of this idea is given in the following.

Lemma 1 *Let ν be any Hider distribution on the nodes of a rooted network Q. Let $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_n)$ be disjoint expanding arc sequences starting at a common node X. Let S_{AB} and S_{BA} be two expanding searches of Q which are the same except that, on reaching X, S_{AB} follows the sequence AB while S_{BA} follows BA. Then*

$$T(S_{BA}, \nu) - T(S_{AB}, \nu) = (\lambda(A) \cdot \lambda(B))(\rho(A) - \rho(B)). \tag{8}$$

So if $\rho(A) > \rho(B)$ it is better (lower search time) to search A before B.

Proof. Note that the times spent searching the trees A and B are respectively $\lambda(A) = \Sigma_i \lambda(a_i)$ and $\lambda(B) = \Sigma_j \lambda(b_j)$. So compared with S_{AB}, a Hider in A will be found in time $\lambda(B)$ later by S_{BA}, a Hider in B will be found $\lambda(A)$ sooner by S_{BA}, and a Hider at any other location will be found at the same time. Since the respective probabilities of the Hider being located in A and B are $\nu(A)$ and $\nu(B)$, we have

$$\begin{aligned} T(S_{BA}, \nu) - T(S_{AB}, \nu) &= \nu(A) \cdot \lambda(B) - \nu(B) \cdot \lambda(A) \\ &= \lambda(A) \cdot \lambda(B) \frac{\nu(A) \cdot \lambda(B) - \nu(B) \cdot \lambda(A)}{\lambda(A) \cdot \lambda(B)} \\ &= (\lambda(A) \cdot \lambda(B)) \left(\frac{\nu(A)}{\lambda(A)} - \frac{\nu(B)}{\lambda(B)} \right). \end{aligned}$$

■

In applying this result, it is useful to observe that for disjoint subsets A and B, the density of $A \cup B$ is a weighted average of that of A and B, that is

$$\begin{aligned} \rho(A \cup B) &= \frac{\nu(A) + \nu(B)}{\lambda(A) + \lambda(B)} = \frac{\lambda(A)}{\lambda(A) + \lambda(B)} \frac{\nu(A)}{\lambda(A)} + \frac{\lambda(B)}{\lambda(A) + \lambda(B)} \frac{\nu(B)}{\lambda(B)} \\ &= \tfrac{\lambda(A)}{\lambda(A)+\lambda(B)} \rho(A) + \tfrac{\lambda(B)}{\lambda(A)+\lambda(B)} \rho(B). \end{aligned}$$

Consequently if $\rho(B) \leq \rho(A)$ then

$$\rho(B) \leq \rho(A \cup B) \leq \rho(A), \text{with strict inequalities if and only if } \rho(B) < \rho(A). \tag{9}$$

3.2 Known Hider distribution on nodes of a tree

When Q is a tree, we naturally orient all the arcs away from its root O and for a node or arc z we will write Q_z to denote the subtree of Q starting at z, containing z and all arcs above z in Q. If x is a branch node of Q adjacent to arcs a and b going away from the root of Q, we call the subtrees Q_a and Q_b *branches* at x. We fix a Hider distribution ν and consider the densities of all subtrees of Q rooted at O. We will be particularly concerned with those subtrees which have maximum density $r = r(Q)$. Generically, there will be a unique subtree of maximum density r, and the main result of this section is that any optimal search must begin with the arcs of this subtree (in some order). The complicating factor is that it is possible there are multiple subtrees of maximum density, which is why Theorem 3 has a more complicated statement.

The set of subtrees $\mathcal{M}$ of maximum density r in Q may be ordered under set inclusion, in which case we call its minimal elements (those without any proper subtree of density r) *min-max subtrees.* The structure of $\mathcal{M}$ is described in the following straightforward lemma, whose proof is left to the reader.

Lemma 2 *Let ν be a measure on the nodes of a rooted tree Q, O. The collection $\mathcal{M}$ of rooted subtrees of Q which have the maximum density $r = r\,(Q)$ is closed under union and intersection (if we include the subtree with no arcs). These subtrees have the following properties:*

(i) *Distinct min-max subtrees are disjoint.*

(ii) *Every branch at O of a maximum density subtree A has density r.*

(iii) *Every min-max subtree A of Q has only one branch at O.*

Since $\mathcal{M}$ is closed under unions, it has a unique maximal element (the union of all its elements), which we call the *max-max subtree and* denote by $M = M\,(Q, \nu)\,.$ Generically, there is a unique subtree of maximum density (namely M). We say that a search S *begins with* the subtree B if some initial block of arcs of S is some ordering of the arcs of B. Generically, there will be a unique subtree of maximum density, and every optimal expanding search must begin with its arcs. In the more general case, the result is as follows.

Theorem 3 *Let ν be any Hider distribution on the nodes of a rooted tree Q. Then*

(i) *Every optimal expanding search of Q begins with some min-max subtree of Q.*

(ii) *Every optimal expanding search of Q begins with the max-max subtree $M = M(Q)$;*

(iii) *For any subtree A of maximum density r, there is an optimal expanding search of Q that begins with the arcs of A.*

Proof. **(i)** We prove the first part of the theorem by induction on the number of arcs: it is trivially true for a network consisting of one arc. So assume that (i) is true for all trees with at most $N-1$ arcs, and let $S=(a_1,a_2,...,a_N)$ be an optimal search of the tree Q having N arcs. Let k be the smallest integer for which tree S_k (consisting of the first k arcs of S) contains some min-max subtree A. Let a_j be the first arc of S contained in A, so that it must be a root arc of A.

We first assume that $j>1$, that S begins with arcs not in A. After searching $S_{j-1}=(a_1,\ldots,a_{j-1})$, the induction hypothesis says S must continue by searching some min-max subtree B of the remaining tree S^{j-1}. Since B contains the root arc a_j of Q, it must also be a min-max subtree of Q. But as both A and B are min-max subtrees of Q starting with the same root arc a_j, Lemma 2 (i) says that $B=A$. Hence S can be written as $S=S_{j-1},A,X$. If we had $\rho\left(S_{j-1}\right)\geq r$, then S_{j-1} would be a tree of maximum density, contradicting the minimality of k, so we must have that $\rho\left(S_{j-1}\right)<r=\rho\left(A\right).$ But this would contradict the optimality of the expanding search S, as Lemma 1 shows that the alternative expanding search $S'=A,S_{j-1},X$ has a strictly lower expected search time. It follows that our assumption $j>1$ is false, and this case is impossible.

So the search S begins with a sequence of arcs $S_l=(a_1,\ldots,a_l)$ belonging to A. Suppose l is the largest integer for which $S_l\subset A$. If $S_l=A$, the proof is complete. If not, by the induction hypothesis applied to the tree S^l, S must continue by searching some min-max subtree B of S^l. Note that B must be disjoint from A, by Lemma 2 (iii). By the maximality of $r(S^l)$, we must have $\rho(B)=r(S^l)\geq\rho(A-S_l)$, and by the minimality of A, $\rho(A)>\rho(S_l)$ so that $\rho(A-S_l)>\rho(S_l)$, by 9. Hence $\rho(B)>\rho(A)$, and $\rho(B\cup A)>\rho(A)=r$, contradicting the maximality of r.

(ii) By part (i), the first arc of S is certainly in M since S begins by searching a min-max subtree, which must be contained in M. Suppose k is the largest integer for which $S_k\subset M$. If $S_k=M$, the proof is complete, so suppose not. Note that by definition of r, we must have $r(S^k)\geq\rho(M-S_k)$, since $M-S_k$ is a rooted subtree of S^k. Also, since $\rho(M)=r$ and $\rho(S_k)\leq r$, by 9, $\rho(M-S_k)\geq r$. Putting these together gives $r(S^k)\geq r$. By part (i), after searching S_k, S must search a min-max subtree, A of S^k with $\rho(A)=r(S^k)\geq r$. The tree A must begin with an arc not in M, and hence be disjoint from M by Lemma 2 (iii), so that $\rho(A\cup M)\geq r$, by (9), contradicting the maximality of M.

(iii) Let ε_n be a positive sequence tending to 0. For each n, define the measure ν_n on the nodes of Q such that $v_n\left(x\right)=\left(1+\varepsilon_n\right)\ \nu\left(x\right)$ for all nodes x in M and $v_n\left(y\right)=\left(1-\omega_n\right)\ \nu\left(y\right)$ for all nodes y not in M, where ω_n is chosen to make ν_n a probability measure. For each n, let S_n be an optimal expanding search on Q with the Hider distribution ν_n. Since there are only a finite number of expanding searches, one of them, call it S', must appear infinitely often in the sequence $\left(S_n\right)_{n=1}^{\infty}$, and hence also be optimal for the limiting distribution ν. But for any of the measures ν_n, the tree A is the unique, and hence maximal, subtree of maximum density, so by part (ii), any optimal expanding search must start with A. Hence in particular the optimal expanding search

S' starts with A. ■

We now have a simple algorithm for constructing an optimal expanding search (indeed all such searches) on a tree when the Hider has a known distribution on the nodes. It is sufficient, after the first k arcs of an optimal expanding search S have been chosen, to determine which initial arcs of the remaining tree S^k lead to optimal expanding searches. The answer is that any arc of the max-max subtree of S^k can be chosen, or equivalently the unique root arc of any min-max subtree of S^k. Summarizing, we have the following.

Corollary 4 *Let ν be a Hider distribution on the nodes of a tree Q and let M be the max-max subtree of Q. Then any arc a at the root of Q which belongs to M can be taken as the first arc a_1 of an optimal expanding search. Repeating this for the subtree Q_1 obtained by deleting a_1 from Q and identifying its tip with the root, gives all possibilities for a_2, and so on.*

We saw in Section 1 how this algorithm can be used to find the optimal search of the network Q_0 of Figure 1 with arc lengths given by (3). We now illustrate how the algorithm can be applied to a network whose maximum density subtree is not unique. Consider the network Q depicted in Figure 1, but this time suppose all the arcs have length 1 and the nodes A, B, C_1, C_2, D_1, and D_2 have respective measures $0.1, 0.2, 0.3, 0.1, 0.2$, and 0.1. The first step of the algorithm is to identify the max-max subtree, $M(Q)$, which can easily be found to be abc_1d_1. The next step is to choose one of the root arcs of $M(Q)$ as the first arc of the search. Depending on whether we choose a or b, we obtain a different new network, Q_1 in which we have shrunk a to the root, or Q_1', in which we have shrunk b to the root, as depicted below.

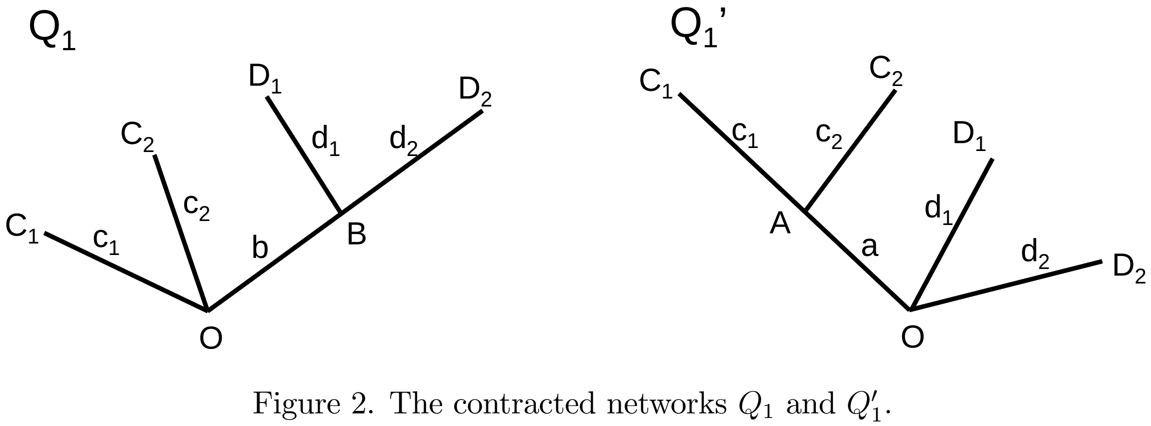

Figure 2. The contracted networks Q_1 and Q_1'.

Suppose we choose to begin with b. Then we find the max-max subtree of Q_1' is ac_1d_1, so that we can choose either a or d_1. Suppose we choose d_1 and shrink this arc to the root. The max-max subtree of the resulting network is ac_1, and we continue by searching these two arcs next. Finally, shrinking these arcs to the root we are left with the network consisting of only the arcs c_2 and d_2. These arcs have equal density so the max-max subtree is the whole network, and c_2 and d_2 can be searched in either order. The result is an optimal search $(b, d_1, a, c_1, c_2, d_2)$ with expected search time equal to $0.2(1) + 0.2(2) + 0.1(3) + 0.3(4) + 0.1(5) + 0.1(6) = 3.2$. Had we made different choices when applying the algorithm we may have produced any one of the alternative optimal strategies: $(b, d_1, a, c_1, d_2, c_2), (b, a, c_1, d_1, c_2, d_2), (b, a, c_1, d_1, d_2, c_2), (a, c_1, b, d_1, c_2, d_2)$, or $(a, c_1, b, d_1, d_2, c_2)$.

It is worth noting that the principle of searching the subtree of highest density first does not hold in the analogous pathwise search problem, where a Searcher must take a continuous path in a rooted network to minimize the time taken to find a Hider located on the network according to some distribution. For example, consider the network consisting of the closed interval $[-2, 2]$ rooted at 0, where the Hider is located at $-2, +1, +2$ with respective probabilities $\frac{7}{13}$, $\frac{4}{13}$ and $\frac{2}{13}$. The unique maximum density subtree is $[0, 1]$, but if a search begins with a path from 0 to 1, the minimum possible expected search time is given by the path which continues to -2 before going to $+2$. This has expected search time $\frac{48}{13}$, whereas a search which begins by going to -2 before going to 1 and then 2 has a smaller expected search time of $\frac{46}{13}$.

3.3 Hider distribution with greatest search time on a tree

Fix a tree Q and define, for any Hider distribution, the minimum search time

$$T(\nu) = \min_S T(S, \nu), \text{ where } S \text{ varies over expanding searches.} \tag{10}$$

We seek the distribution ν which maximizes $T(\nu)$. Clearly, by domination, we need only consider those distributions ν concentrated on the leaf nodes of Q, and these form a compact set (a simplex in fact). For any fixed S, the expected search time $T(S, \nu)$ is continuous in ν, and hence the minimum function $T(\nu)$ over all such S is continuous. So a maximizing distribution ν exists. Here we show that it is unique and that it is the EBD distribution, well known in the theory of pathwise search.

Definition 5 *On any rooted tree Q, the Equal Branch Density (EBD) distribution is the measure e on the leaf nodes of Q such that at every branch node x all the branches have the same density with respect to e. (A formula for this unique measure is given in Alpern (2011a) and a recursive method of calculating it is given in Gal (1979)).*

For example, suppose we wish to assign the EBD distribution to the tree Q_0 of Figure 1 with arc lengths given in (3). To ensure that the subtrees ac_1c_2 and bd_1d_2 have the same density we put $e(ac_1c_2) = \frac{1}{5}$, and $e(bd_1d_2) = \frac{4}{5}$, giving $\rho(ac_1c_2) = \rho(bd_1d_2) = \frac{1}{15}$. Since $c_1 = c_2$ and $d_1 = d_2$, the measure on each of these subtrees is divided equally between their two nodes, so that $e(C_1) = e(C_2) = \frac{1}{10}$ and $e(D_1) = e(D_2) = \frac{2}{5}$. As we noted at the end of Section 1, the EBD measure is the optimal Hider distribution in the expanding search game (4) played on Q_0.

We begin by characterizing the EBD measure in terms of our new notion of the max-max tree.

Lemma 6 *The EBD distribution e on a rooted tree Q is the unique Hider distribution v such that for each node x, the max-max subtree of Q_x is Q_x itself.*

Proof. If for any branch node x, the max-max subtree of Q_x is Q_x itself, then by Lemma 2 (ii) all the branches of Q_x must have the same density $r(Q_x)$, so the distribution must be e.

To show that e does indeed have this property, suppose the Hider measure on the leaf nodes is e and there exists a node x for which the max-max subtree of Q_x is some proper subtree $A \subset Q_x$. Then we can find some node y in Q_x with outward arcs a and b such that $Q_a \subset A$ and Q_b is disjoint from A. But $\rho(Q_a) \geq r(Q_x)$, since otherwise A would not be of maximum density, and by the definition of e, Q_a and Q_b have equal density, so that $\rho(A \cup Q_b) \geq r(Q_x)$, contradicting the maximality of A. Hence the max-max subtree of every subtree Q_x is Q_x. ■

Lemma 7 *If ν is a measure on a tree Q with maximum search time $T(\nu)$, as defined in (10), then the max-max subtree with respect to ν, $M(Q_x, \nu)$ of each subtree Q_x is Q_x itself.*

Proof. Suppose ν has maximum search time but there is some x_0 with $M(Q_{x_0}) \neq Q_{x_0}$. Clearly ν must be supported on the leaf nodes of Q. Suppose y and z are leaf nodes with $y \in M(Q_{x_0})$ and $z \notin M(Q_{x_0})$. Suppose $S = (a_1, a_2, ..., a_N)$ is an optimal search, and the k is the smallest integer for which S_k is disjoint from Q_{x_0}. Then by Theorem 3 (ii), Q_{x_0} contains a branch, A_0 of the max-max subtree $M(S^k)$, with $\rho(A_0) = r(S^k)$. By definition of r, $r(Q_{x_0}) \leq r(S^k)$, so by the maximality of $M(S^k)$ we must have $A_0 = M(Q_{x_0})$, so that $y \in M(S^k)$ and $z \notin M(S^k)$. Hence every optimal expanding search S on Q, ν must search y before z. Let ε_n be a positive sequence bounded above by $\nu(y)$ and tending to 0. For every n, define ν_n to be the measure on Q which agrees with ν except that $\nu_n(y) = \nu(y) - \varepsilon_n$ and $\nu_n(z) = \nu(z) + \varepsilon_n$. Let S_n be an optimal expanding search on Q, ν_n, and let S' be one which appears infinitely often in the sequence $(S_n)_{n=1}^{\infty}$. Thus S' is optimal for a subsequence of the measures ν_n and hence by continuity S' is optimal for the limit sequence ν. Let v_m be one of the measures for which S' is optimal. Clearly $T(S', \nu_m) - T(S', \nu) = \varepsilon_m (T(S', z) - T(S', y)) > 0$. But as S' is optimal for both measures this implies that $T(\nu_m) = T(S', v_m) > T(S', \nu) = T(\nu)$, contradicting the assumed maximality of the search time of ν. ■

Theorem 8 *On any tree Q, the EBD distribution e is the unique Hider distribution which maximizes the minimum search time $T(\nu)$ over all distributions ν.*

Proof. The distribution ν varies over a compact k-simplex, where k is the number of leaf nodes. The function $T(\nu) = \min_S T(S,\nu)$ over the finite number of expanding searches is continuous in ν, so a maximizing distribution $\bar{\nu}$ exists. By Lemma 7 we must have $M(Q_x, \bar{\nu}) = Q_x$ for every node x and then by Lemma 6 we have $\bar{\nu} = e$. ■

Corollary 9 *On any tree Q, the maximum search time $T(e)$ is given by $T(e) = \frac{1}{2}(\mu + D)$, where $D = D(Q)$ is the mean distance from the root to the leaf nodes, with respect to the EBD measure e on those nodes. That is,*

$$D = \sum_{\textit{leaf nodes } x} e(x) \cdot d(O,x).$$

Proof. If Q has a single arc then $T(e) = \mu$ and $D = \mu$, so the result is true for such trees. So assume result is true for all trees with fewer arcs than Q, and assume there are two arcs a and b at the root, leading to subtrees Q_a and Q_b. First note that if μ_x denotes the length of the subtree Q_x, by the definition of e we have that

$$\frac{\mu_a}{\mu} D(Q_a) + \frac{\mu_b}{\mu} D(Q_b) = D. \tag{11}$$

Since Q_a is a maximum density subtree, by Theorem 3 (iii) there is an optimal search S_{ab} which begins with a search of Q_a and continues with a search S_b of Q_b, and so

$$T(S_{ab}, Q) = \frac{\mu_a}{\mu} T(S_a, Q_a) + \frac{\mu_a}{\mu}(\mu_a + T(S_b, Q_b)). \tag{12}$$

Since all searches of Q_a have maximum time μ_a and S_{ab} is optimal, it must also be true that S_a and S_b are optimal on their respective subtrees. So by the induction hypothesis

$$T(S_x, Q_x) = \frac{\mu_x + D(Q_x)}{2}, \ x = a, b, \tag{13}$$

where D is taken with respect to the normalized EBD distribution e_x on Q_x, $e_x = (\mu/\mu_x)\, e$. Combining (12),(11), and (13), we have

$$\begin{aligned} T(S,Q) &= \frac{\mu_a}{\mu} \cdot \frac{\mu_a + D(Q_a)}{2} + \frac{\mu_b}{\mu} \cdot \left(\mu_a + \frac{\mu_b + D(Q_b)}{2}\right) \\ &= \frac{\mu_a^2 + 2\mu_a\mu_b + \mu_b^2}{2\mu} + \frac{\mu_a D(Q_a) + \mu_b D(Q_b)}{2\mu} = \frac{\mu + D}{2}. \end{aligned}$$

If the root has more than two arcs, the same proof works with Q_b denoting the subtree $Q - Q_a$. If the root has one arc a then let Q' be the tree $Q - a$ with root at the end of a, and observe that

$$\begin{aligned} T(e) &= \mu_a + T_{Q'}(e') = \mu_a + \frac{\mu(Q') + D(Q')}{2} \\ &= \mu_a + \frac{(\mu - \lambda(a)) + (D(Q) - \lambda(a))}{2} = \frac{\mu + D}{2}. \end{aligned}$$

■

For an example of the calculation of D, see the end of Section 1, where we determine $D(Q_0)$, for the network Q_0 depicted in Figure 1 with arc lengths given by (3).

3.4 A solution method for Hider on nodes of arbitrary network

We noted above that every search $S = (a_1, \ldots, a_N)$ on a network Q determines a spanning subtree S_N of Q. If S is optimal against Q, ν, it must also be optimal (the optimal ordering of the arcs) on that spanning subtree. So a method to solve the Bayesian search problem on any network Q is to identify all spanning trees, use Corollary 4 to find the minimum search time in each case, and take the minimum of these.

We illustrate this method by considering the cycle network shown below in Figure 3, where the numbers on the nodes represent the Hider probability ν, in 1/20'ths. Suppose $\lambda(a) = 2$, but the other arcs have unit length. We will determine the minimum search time $T(\nu)$ for the Hider distribution ν, using the algorithm described above. There are four spanning subtrees, each specified by which of the four arcs of Q is avoided. If a is avoided, we get the unique search (d, c, b), with expected search time $(1 \cdot 1 + 2 \cdot 7 + 3 \cdot 12)/12 = 51/20$. If b is avoided, the max-max tree is a, so the unique optimal search of $Q - b$ is (a, d, c), with expected search time $(2 \cdot 12 + 3 \cdot 1 + 4 \cdot 7)/20 = 55/20$. If c or d are avoided, the max-max tree is ab, and the respective optimal searches are (a, b, d) and (a, b, c) with common expected search time of $(2 \cdot 12 + 3 \cdot 7 + 4 \cdot 1)/20 =$

49/20, which is the global optimum.

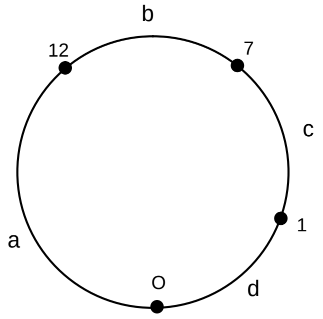

Figure 3. A cycle network W.

3.5 A counterexample to max-max subtree M first

We showed in part (ii) of Theorem 3 that the max-max subtree of a tree network Q must be searched first by any optimal expanding search. The concept of the max-max subtree, as a maximal subtree of maximum density, can be defined on any network. A natural conjecture is that an optimal expanding search on any network must begin with the (generically unique) max-max tree. However it turns out that this conjecture can fail even for the simplest non-tree network, the cycle network.

Consider the particular cycle network of the previous subsection. It is easy to calculate that the max-max tree is dcb, with density $20/3 \approx 6.666\,7$. If the conjecture were true, the optimal expanding search would have to be (d, c, b), but we have seen that this search gives a suboptimal expected search time of 51/20.

4 Known Continuous Hider Distribution

In the previous section we analyzed the Bayesian search problem of minimizing the expected time to find a Hider hidden according to a known distribution ν on the nodes of a network Q. The assumption that the Hider is at a node greatly simplified the analysis - in particular the finiteness of the set of expanding searches made the existence of optimal searches obvious. We now relax the assumption that the Hider must be at a node of Q and allow him any distribution ν on the whole network Q. This in turn requires searches that might go a certain distance along one arc and then switch to searching another one before the end of the first arc is reached. In some rare (non-generic) cases, the Searcher might even have to go along two arcs *at the same time* (but with speeds summing to 1).

4.1 General expanding searches

The idea behind our general definition of an expanding search S concerns the closed region $S(t)$ of Q that it has searched by time t. As t increases, its size (total length) cannot grow too fast and it cannot jump to new points detached from those it has already searched. A Hider at H is captured when H first belongs to the searched set $S(t)$. These ideas are captured in the following definition.

Definition 10 *An expanding search S on a network Q is a nested family of connected closed sets $S(t)$, $0 \leq t \leq \mu$, which satisfy*

(i) $S(0) = O$ *(starts at the root of Q), $S(\mu) = Q$ (exhaustive search),*

(ii) $S(t') \subset S(t)$ *for $t' < t$, and*

(iii) $\lambda(S(t)) = t$.

The set of all expanding searches is denoted by $\mathcal{S}$.

It follows from (ii) and (iii) that $\lambda(S(t) - S(t')) = t - t'$ and consequently by connectedness that $d_{Haus}(S(t), S(t')) \leq t - t'$, where d_{Haus} is the Hausdorff metric on Q. The set $\mathcal{S}$ is compact in the uniform Hausdorff metric

$$d^*(S, S') = \max_{0 \leq t \leq \mu} d_{Haus}(S(t), S'(t)). \tag{14}$$

For $S \in \mathcal{S}$ and $H \in Q$ let $T^* = \inf\{t : H \in S(t)\}$. Suppose $H \notin S(T^*)$. Then for $T^* < t < T^* + d_{Haus}(S(t), S(T^*))$ we also have that $H \notin S(t)$, contradicting the definition of T^*. Consequently $H \in S(T^*)$. This shows that the infimum is a minimum, and leads to the definition

$$T(S, H) = \min\{t : H \in S(t)\}.$$

Next we show that for any fixed $H \in Q$, $T(S, H)$ is lower semicontinuous for $S \in \mathcal{S}$ with respect to the uniform Hausdorff metric d^*. Suppose $T(S, H) > t_0$. Then $H \notin S(t_0)$ and furthermore if $d^*(S, S') < d(H, S(t_0))$ it follows that $H \in S'(t_0)$ and so $T(S', H) > t_0$. Thus $T(S, H)$ is lower semicontinuous in S for fixed H and hence also $T(S, h) = \int_Q T(S, H)\ d\nu(H)$ is lower semicontinuous for any fixed Hider distribution ν on Q. It follows that for any fixed ν there is always an expanding search S which is optimal in that it minimizes the expected capture time $T(S, h)$. The optimal strategy need not be unique. Similar arguments will show that the zero sum game where the Searcher picks $S \in \mathcal{S}$ and the Hider picks $H \in Q$ has a value V. To summarize, we have shown the following.

Theorem 11 *Given any Hider distribution ν on a network Q, there exists an expanding search $S \in \mathcal{S}$ which minimizes the expected capture time $T(S, \nu)$.*

4.2 Pointwise search

For an expanding search it is not in general possible to say where the Searcher is located at a particular time t, unlike for example in the pathwise paradigm in Gal's analysis of search games, where the Searcher simply follows a continuous path in Q. Of course if $S([0,t]) - S([0,t))$ is a single point $P(t)$, this is where the Searcher would be located at time t. In fact we can characterize functions $P(t)$ which arise in this manner as *pointwise searches.*

Definition 12 *A* ***pointwise search*** *of a network* (Q, A) , *is a surjective map* $P : [0, \mu] \to Q$ *with* $P(0) = O$ *(root), such that for all* $t \in (0, \mu]$ *we have*

$$d(P(t'), P(t)) \leq t - t' \text{ for some } t' < t. \tag{15}$$

The set of all pointwise searches is denoted by $\mathcal{S}^p$.

In other words for $P \in \mathcal{S}^p$, the Searcher position $P(t)$ at any time t is connected to some previously visited point $P(t')$ by a continuous path. Note that the usual paradigm of pathwise search in search games has the same definition as (15) except that the quantifier *some* is replaced by *all*. That is, $P(t)$ is connected by a path to the root $P(0)$ in pathwise search. Unlike pathwise searches, pointwise searches are not in general continuous. (This is also true in discrete path planning algorithms, where successive nodes need not be adjacent, but each node is ajacent to a previous node.) There are certain *restart times* $\mathcal{T}_S = \{t : P(t) = P(t'),\ \text{some } t' < t\}$ when the search returns to *restart points* $P(\mathcal{T}_P)$ which it has visited earlier. In the special case of combinatorial searches (expanding arc sequences) considered in Section 3, the restart points are those nodes which are tails of two or more arcs involved in the search. The fact that intervals (arc interiors) of zero density are searched without interruption (that is, full arcs are searched) is obvious. This result also applies to arcs where the distribution is uniform. This is a version of Lemma 1 - see Corollary 8 of Alpern and Howard (2000).

Given a pointwise search $P \in \mathcal{S}^p$ and a hiding point $H \in Q$, the capture time $T = T(P, H)$ is given by

$$T(P, H) = \min\{t : P(t) = H\}.$$

Furthermore the set $\mathcal{S}^p$ of pointwise searches is dense in $\mathcal{S}$ in the uniform Hausdorff topology given by (14). To see this, fix any search $S \in \mathcal{S}$ and for a positive integer n let $t_i = i\mu/n$ for $i = 0, \ldots, n$. Let $P \in \mathcal{S}^p$ be a search such that $P([0, t_i]) = S(t_i)$, $i = 0, \ldots, n$. Such a P is easily constructed, as the subarcs in $S(t_i) - S(t_{i-1})$ can be traversed one at a time by P in the interval $[t_{i-1}, t_i]$. A similar argument in a slightly simpler setting is worked out in detail in Lemma 2 of Alpern and Howard (2000). The significance of the fact that pointwise searches are dense is that we can obtain $\varepsilon-$optimal expected search strategies using them against any Hider distribution.

An example of a Q, ν where the minimum search time requires a general expanding search rather than any pointwise search is the following.

Example 13 *Let Q be the interval $[-1, 1]$ with its root O at the center 0. Let ν be given by the density function $h(x) = 1 - |x|$. The unique optimal expanding search $S(t)$ is given by $S(t) = [-t/2, t/2]$ for $0 \leq t \leq 2 = \mu$. (A derivation of this fact is given in the next subsection.) Clearly the expanding search S is not induced by any pointwise search P, as $S([0,t]) - S([0,t)) = \{-t/2, t/2\}$ is not a singleton set.*

4.3 Search on a tree again

We now return to the problem of searching against a known Hider distribution on a tree, which we solved in Section 3.2 for the case where the distribution was concentrated on nodes. Here we consider the general case. In that section we were able to restrict our attention to combinatorial subtrees (e.g. connected sets of arcs including an arc at the root). Here we use the term subtree to denote any closed connected subset of the tree Q which contains the root O. As before, we will be concerned with subtrees of maximum density, but now such a subtree may not exist. For instance for Q, ν of Example 13. However when such a subtree exists, we can prove the continuous analog of Theorem 3.

Theorem 14 *Let the Hider H be hidden according to a known distribution ν on a rooted tree Q and suppose there is a unique rooted subtree A of maximum density. Then there is an optimal expanding search S which begins by searching A. That is, $S(\lambda(A)) = A$.*

It turns out that this result may be reduced to the very simple case where Q is a star with just two arcs a_1 and a_2, after which we may apply the solution of the alternating search problem given in Alpern and Howard (2000). On arc i the probability with respect to ν that the Hider is within distance t of the root is denoted by F_i. It is clear that an expanding search in this context is determined by a nondecreasing continuous *alternation* function $\alpha(t)$, $0 \leq t \leq \lambda(a_1) + \lambda(a_2)$, with $\alpha(t) - \alpha(t') \leq t - t'$ for $t' < t$, where $S_\alpha(t)$ is the subtree going distance $\alpha(t)$ on a_1 and $t - \alpha(t)$ on arc a_2. The expected capture time associated with S_α is

$$\begin{aligned} T(\alpha) &= \int_0^{\lambda_1 + \lambda_2} t \, dF^\alpha(t), \text{ where} \\ F^\alpha(t) &= F_1(\alpha(t)) + F_2(t - \alpha(t)). \end{aligned}$$

The α minimizing $T(\alpha)$ for given functions F_i, is called the *optimal interleaving of the two distributions F_1 and F_2*. Often *simple* alternation strategies, those where α' is always 0 or 1, are optimal. The theory of optimal interleaving of two distributions is given in Section 5 of Alpern and Howard (2000), where the following is proved (a special case of Corollary 6 of that paper).

Proposition 15 *Suppose that each ratio $F_i(t)/t$ has a maximum of r_i at t_i, and that $r_1 > r_2$. Then any optimal alternation strategy satisfies $\alpha(t_1) = t_1$, that is, it begins by a search on a_1.*

We now use the ideas of optimal interleaving to prove Theorem 14.

Proof. Let S be an optimal expanding search strategy, which must exist by Theorem 11. Let t_0 be the smallest t for which $A \subset S(t)$ and define $B = S(t_0) - A$. For $t \leq t_0$, let $f(t)$ be the earliest time that the tree A has been searched to length t, that is, $f(t) = \min\{y : \lambda(S(y) \cap A) = t\}$. It follows that $t - f(t) = \lambda(S(t) \cap B)$. Similarly let $g(t) = \min\{y : \lambda(S(y) \cap B) = t\}$. Let S^* be the expanding search which searches A and then B, each in the same relative order as in S, that is,

$$S^*(t) = \begin{cases} S(f(t)) \cap A & \text{for} \quad 0 \leq t \leq \lambda(A), \\ A \cup (S(g(t - \lambda(A))) \cap B) & \text{for} \quad \lambda(A) \leq t \leq \lambda(A) + \lambda(B) = t_0. \end{cases}$$

According to Proposition 15, $S^*(t)$ is an optimal interleaving of the distributions $S(t) \cap A$ and $S(t) \cap B$, for $t \leq t_0$, so the expected capture time for S^* is not larger than that for S. Consequently, as $S^*(t)$ is also an expanding search, it must also be optimal. ■

Unlike the problem in which ν is concentrated on nodes, there now may be no tree of maximum density, since $\{\rho(A) : A \neq \phi$ is a rooted subtree of $Q\}$ may not have a supremum. For instance in Example 13 the subtrees $[0, x]$ or $[-x, x]$ have density $1 - x/2$, with limiting density 1. But clearly no subtree has density as high as 1. So we cannot apply (at the moment) Theorem 14. However there is a simple trick we can use, based on the simple observation that on any network that begins with a single arc a at the root, we must begin by searching a. So we reverse engineer this idea by considering the network $Q + L$ in which a new arc of length $L > 0$ is added at the root of Q, and its other end O' is taken as the root of $Q + L$. We draw Q and $Q + L$ below in Figure 4 for the interval network Q of Example 13.

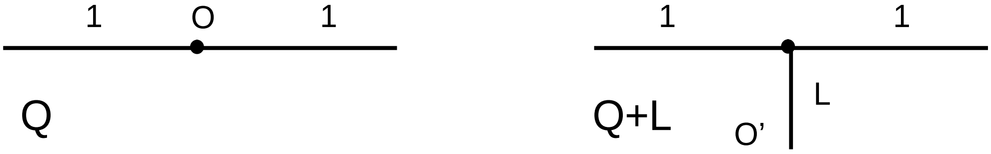

Figure 4. The network $Q + L$.

It is easy to calculate that the unique maximum density subtree of $Q + L$ intersects Q in some subtree $M(L) = [-x(L), x(L)]$ for which the density

$$\frac{\nu([-x, x])}{2x + L} = \frac{x(2 - x)}{2x + L} \text{ is maximized,}$$

giving $x = x(L) = \left(\sqrt{L(L+4)} - L\right)/2$. Thus Theorem 14 says that an optimal expanding search of $Q + L$ must begin with the new arc and then the subtree $M(L)$ of Q, and hence any optimal expanding search of Q must begin with the subtree $M(L)$. Thus by increasing L, we generate an increasing family of initial subtrees of an optimal search.

In general, note that by adding an arc a of length L to a rooted tree Q to obtain $Q+L$, we remove the problem that there may not be a maximum density subtree, since the set $\{\rho(A) : A \neq \phi \text{ is a rooted subtree of } Q+L\}$ is equal to $\{\rho(B \cup a) : B \text{ is a rooted subtree of } Q\}$, which is the upper-semicontinous image of a compact set with respect to the Hausdorff metric, so it must have a maximum. In the generic case, there is a unique maximum density subtree, $M(L) \cup a$ of $Q + L$. It can be easily established that the subtrees $M(L)$ are nested, so that if $L_1 < L_2$ we have $M(L_1) \subset M(L_2)$.

This approach can also be applied to find the optimal search (arc sequence) for a Hider distributed on the nodes of a tree. In this case, $M(L)$ will 'skip' a finite number of times as L increases from 0 to ∞. We illustrate these ideas using the example of the network Q drawn in Figure 1, with arc lengths and Hider distribution as depicted in Figure 5 below.

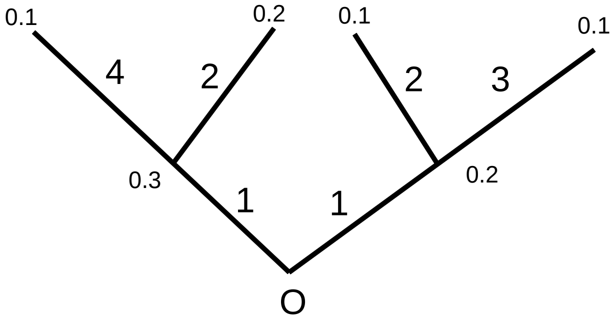

Figure 5. The network Q.

It is easy to see from Corollary 4 that the optimal search strategy is $(a, b, c_2, d_1, d_2, c_1)$. To illustrate the $Q + L$ solution method, we plot the graphs of the density functions of the five rooted subtrees of Q which have maximum density in $Q + L$ for some L. For example, the function plotted for tree a is $0.3/(1+L)$. As L increases from 0, the highest densities in $Q + L$ are the subtrees determined by a, ab, abc_2, abc_2d_1, $abc_2d_1d_2$ and finally Q itself (not shown in the figure).

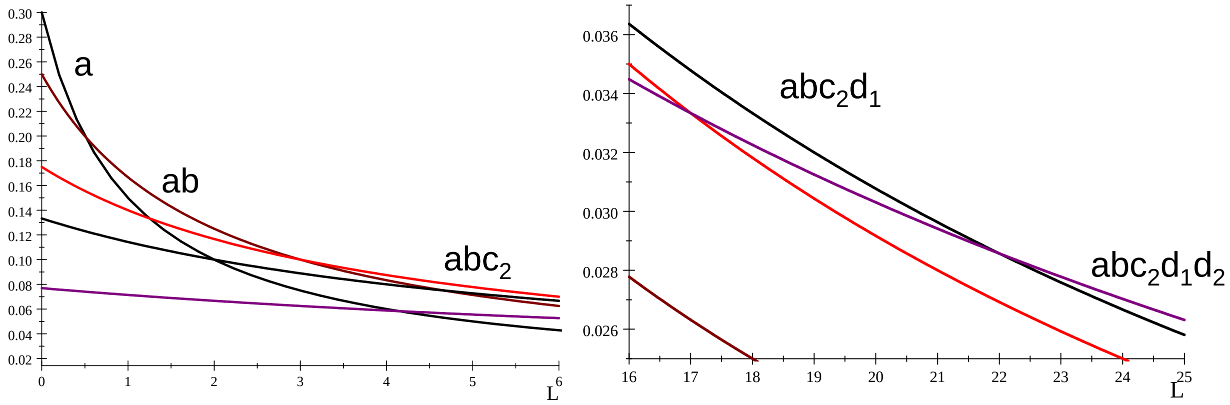

Figure 6. Densities of some rooted subtrees of Q in the tree $Q\,(L)$

The $Q+L$ method does not always work so smoothly as it is possible that the subtree sequence $M\,(L)$ may increase by more than a single arc at the time, in which case the order of those arcs would have to be obtained by the method given in Section 3.2. An example of this behaviour can be seen in the following network, all of whose arcs have length 1.

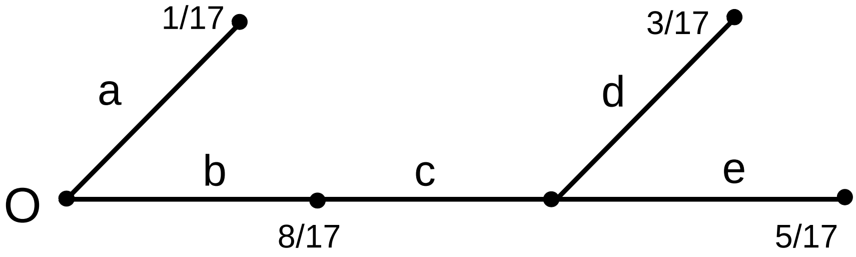

Figure 7. A network where $M(L)$ skips by more than one arc.

When $L < 2$, $M(L) = b$, indicating that the first arc searched should be b. But for $2 \leq L < 12$, $M(L) = bcde$, so although the method determines that c should be searched next, followed by d and e, it does not indicate in which order d and e should be searched. We therefore need to use Theorem 3 which says that since e is the maximum density subtree of de, this arc must be searched next. When $L \geq 12$, $M(L)$ is the whole network $abcde$, and a should be searched last, which is clear as it is the only remaining arc.

5 Unknown Hider Distribution - Search Games

We now assume that the Hider distribution ν is not known to the Searcher. In this case we consider the problem of finding the mixed Searcher strategy (probability measure over expanding searches) which minimizes the expected search time in the worst case. An equivalent problem, which we prefer to adopt, is the zero-sum Expanding Search Game $\Gamma(Q)$. Here the maximizing Hider picks a location H in Q, the minimizing Searcher picks an expanding search S in $\mathcal{S}$, and the payoff is the search (capture) time $T(S, H)$. We will see that all the searches used in this section will be combinatorial, in the sense defined at the beginning of Section 3, so that they consist of expanding arc sequences. In Section 4.1 we showed that $\mathcal{S}$ is compact in the uniform Hausdorff metric (14) and that T is lower semicontinuous in S for fixed H. So the Minimax Theorem of Alpern and Gal (1988) gives the following.

Theorem 16 *For any network the expanding search game $\Gamma(Q)$ has a value $V = V(Q)$, the Searcher has an optimal mixed strategy and the Hider has ε-optimal mixed strategies.*

For comparison with a much studied class of games, let $\Gamma^p(Q)$ and V^p denote the pathwise search game and its value, where the Searcher is restricted to pathwise searches. Clearly $V \leq V^p$.

Two simple cases are when Q is a ray from the root and when Q is a circle. In the former case the Hider hides at the leaf node, the Searcher goes there, and the value is the length μ of this ray. In the latter case, the Hider can either hide at the antipodal point to the root or uniformly, the Searcher goes around the circle in a random direction, and the value is $\mu/2$, where μ is the circumference. As in pathwise search, the analysis for the circle goes over to Eulerian networks. However we will show that, unlike the situation for pathwise search, the result for the single ray does not go over to general trees.

A Hider distribution which is available on any network is the *uniform distribution* u which is simply the normalized version of the length, that is, $u(Z) = \lambda(Z)/\lambda(Q) = \lambda(Z)/\mu$ for any measurable set Z. We observed in the previous paragraph that u is an optimal Hider strategy on a circle. Since the definition of expanding search includes the condition $\lambda(S(t)) = t$, the cumulative distribution, F of search times against u satisfies $F(t) = u(S(t)) = \lambda(S(t))/\mu = t/\mu$, and hence we have:

Lemma 17 *The uniform distribution u on any network Q satisfies $T(S, u) = \int_0^\mu (t/\mu)\ dt = \mu/2$, for any $S \in \mathcal{S}$, and hence $V(Q) \geq \mu/2$.*

5.1 Search game on a tree

For rooted trees Q, we showed earlier (Theorem 8 and Corollary 9) that the *EBD* distribution had the maximum search time of $(\mu + D)/2$. Since the minimax theorem holds for expanding search games (Theorem 16), this give the following.

Theorem 18 *For the expanding search game on a rooted tree Q, the EBD distribution is the optimal strategy for the Hider and the value is given by $V = (\mu + D)/2$.*

Except for the trivial case where Q is a single arc from the root to the leaf node, we have $D < \mu$, and hence the value V is strictly less than the value $V^p = \mu$ for pathwise search games on trees established by Gal (1979). While the optimal Hider distribution is the same, the optimal pathwise strategy of following a Chinese Postman Tour in a random direction is no longer optimal in general.

For ease of presentation, in the section we restrict our attention to binary trees, that is to say trees which have two out-arcs (away from the root) from every non-leaf node. An optimal Searcher strategy for binary trees can be given in terms of a branching function and corresponding branching strategy. A branching function β assigns to every branch (non leaf) node a probability distribution $(\beta(a), \beta(b))$ over its two out-arcs. After reaching any node x, the branching strategy σ_β chooses the next arc as follows. Let y be the most recently reached node (possibly x) at which there is an untraversed arc. If there is only one untraversed arc at y, take it. Otherwise, choose each arc with its β probability. Recall from Section 3.2 the definition of Q_z, the subtree of Q containing the node or arc, z and all arcs above z. Let $\bar{\beta}$ be the branching function given by 16.

$$\bar{\beta}(a) = \frac{1}{2} + \frac{1}{2\mu_x}(D_a - D_b), \tag{16}$$

where $\mu_x = \lambda(Q_x)$, and $D_a = D(Q_a)$, $D_b = D(Q_b)$. Note that $\overline{\beta}(a)$ is indeed a probability, since $|D_a - D_b| \leq D_a + D_b \leq \mu_a + \mu_b = \mu_x$.

Theorem 19 *The branching strategy $\sigma_{\bar{\beta}}$ reaches every leaf node i of the rooted tree Q in expected time $T(\sigma_{\bar{\beta}}, i) = (\mu + D)/2$. Hence by Theorem 18 it is an optimal strategy for the Searcher.*

Proof. We prove the theorem by induction on the number of arcs. If the tree Q has a single arc of length l with leaf node i, then clearly $T(\sigma_{\bar{\beta}}, i) = l$. But $\frac{1}{2}(\mu + D(Q)) = \frac{1}{2}(l + l) = l$. Assume the result is true for all proper subtrees of Q and let a and b be the two out-arcs of O. Then by the induction hypothesis, *twice* the expected search time, $2T(\sigma_{\bar{\beta}}, i)$ to reach a leaf node i in, for example Q_a, is given by

$$\bar{\beta}(a)(\mu_a + D_a) + \bar{\beta}(b)(2\mu_a + (\mu_b + D_b))$$

By definition of $\bar{\beta}$ this is equal to

$$\begin{aligned} & \left(\frac{1}{2} + \frac{1}{2\mu}(D_a - D_b)\right)(\mu_a + D_a) + \left(\frac{1}{2} + \frac{1}{2\mu}(D_a - D_b)\right)(2\mu_a + (\mu_b + D_b)) \\ = \; & \mu + \left(\frac{\mu_a D_a + \mu_b D_b}{\mu}\right) = \mu + D \text{ (by (11))}. \end{aligned}$$

■

For example, in the game played on the tree Q_0 depicted in Figure 1, with arc lengths given by (3), we have $D_a = 2$ and $D_b = 7$, so that $\bar{\beta}(a) = \frac{1}{2} + \frac{1}{2.15}(2-7) = \frac{2}{3}$. Clearly $\bar{\beta}(c_1) = \bar{\beta}(c_2) = \frac{1}{2}$ and $\bar{\beta}(d_1) = \bar{\beta}(d_2) = \frac{1}{2}$, so the Searcher has an optimal strategy which begins with a with probability $\frac{2}{3}$, then searches c_1 or c_2 equiprobably, followed by b and then d_1 or d_2 equiprobably; and with probability $\frac{1}{3}$ begins with b then searches d_1 or d_2 equiprobably, followed by a and then c_1 or c_2 equiprobably. This is equivalent to the strategy we found to be optimal by numerical methods in Section 1.

For trees, the expanding search game can be viewed as a pathwise game on a tree with asymmetric arc travel times which are zero going towards the root, as long as the Searcher is restricted to depth-first searches. So another way of obtaining the results of this section is via the theory of search games on variable speed networks, as developed in Alpern (2010) and Alpern and Lidbetter (2010). Such methods will not however work for networks with cycles, as there is no similar orientation of the arcs.

Under this interpretation one might expect that, as with the pathwise search game on trees with symmetric travel times, a Random Chinese Postman Tour strategy may be optimal for the Searcher. However, this is not the case. Suppose we take a Chinese Postman Tour $S = (a, c_1, c_1^*, c_2, c_2^*, a^*, b, d_1, d_1^*, d_2, d_2^*, b^*)$ on the tree Q_0 in Figure 1, with arc lengths given by 3, writing a^*, for example as the arc a traversed in its backwards direction in time 0. In the notation of expanding search, S would be written $(a, c_1, c_2, b, d_1, d_2)$. The reverse of S is $S^* = (b, d_2, d_2^*, d_1, d_1^*, b^*, a, c_2, c_2^*, c_1, c_1^*, a^*)$. If we take a weighted average, $\alpha S + (1-\alpha)S^*$ $(0 \leq \alpha \leq 1)$ of these two searches we obtain a Random Chinese Postman Tour. We then have $T(\alpha S + (1-\alpha)S^*, D_1) = \alpha \cdot 10 + (1-\alpha) \cdot 12 = 12 - 2\alpha$ and $T(\alpha S + (1-\alpha)S^*, D_2) = \alpha \cdot 15 + (1-\alpha) \cdot 7 = 8\alpha + 7$. Hence if $\alpha \leq \frac{1}{2}$, $T(\alpha S + (1-\alpha)S^*, D_1) \geq 11$ and if $\alpha \geq \frac{1}{2}$, $T(\alpha S + (1-\alpha)S^*, D_2) \geq 11$, so there is no Random Chinese Postman Tour strategy which ensures that the Searcher will reach both D_1 and D_2 in expected time less than $V(Q_0) = 10.5$. It is easy to show that no other Random Chinese Postman Tour on Q_0 is optimal either.

5.2 Searching in boxes as expanding search on a star network

We may apply Theorem 19 to a special case of the following well-known problem of searching for an object hidden in one of n boxes $i = 1, ..., n$. The Hider picks a box to hide in and the Searcher searches the boxes in any order. Each box i has search cost c_i and if the Hider has chosen that box, then when the Searcher examines it the search ends with probability q_i. The Searcher seeks to minimize, and the Hider to maximize, the total cost of the search. In Roberts and Gittins (1978) and Gittins and Roberts (1979), this game is analyzed in the case where for each i, $c_i = 1$ and $0 < q_i < 1$. A summary of this analysis is also presented in Gittins (1989). We consider the case where there is no overlook probability, so that all the $q_i = 1$, and where the c_i can take any positive value. It is easy to see that this game is equivalent to the expanding search game on the $n-$star tree Q in which the length $\lambda(a_i)$ of arc a_i is c_i. So its total length μ is given by $C = \Sigma_{i=1}^n c_i$. In the EBD distribution e, the Hider chooses the leaf nodes of a_i with probability $e(a_i) = \lambda(a_i)/\mu = c_i/C$, which is the optimal hiding distribution. Furthermore we have

$$\begin{aligned} D &= \Sigma_{i=1}^n e(a_i)\lambda(a_i) = \Sigma_{i=1}^n \frac{c_i^2}{C}, \text{ and so by Theorem 18,} \\ V &= \frac{1}{2}(D+\mu) = \frac{1}{2}\left(C + \frac{\Sigma_{i=1}^n c_i^2}{C}\right). \end{aligned}$$

There are many optimal search strategies. They may be obtained by converting the star into an equivalent binary tree by adding arcs of length zero. For example, suppose the binary tree begins with the arc a_1 and a new arc of length zero leading to the rest of the tree. Then by Theorem 20 the Search branching strategy takes a_1 first with probability

$$\frac{1}{2} + \frac{1}{2C}\left(c_1 - \Sigma_{i=2}^n \frac{c_i^2}{C - c_1}\right)$$

Otherwise he searches a_1 last and the rest of the tree in some order determined by successive branching.

5.3 2-arc connected networks

We showed in the introduction how the expanding search game was easy to solve on the single arc network (with root at one end) and on the circle network. The previous section generalized the first observation to a solution for all trees. Here we generalize the second observation to all networks which remain connected on removal of any (open) arc, the so called *2-arc connected* networks. We will need the following characterization of such networks due to Robbins (1939) in terms of *orientable networks*, those for which the arcs can be oriented in such a way so that there is a directed path from any point x to any other point y in Q.

Theorem 20 *The following are equivalent for a network Q :*

(i) *Q is 2-arc connected.*

(ii) *Q is orientable.*

(iii) *There is an increasing sequence of subnetworks $Q_0, Q_1, \ldots, Q_k = Q$, such that Q_0 is a cycle and each Q_i, $i > 0$, is obtained from Q_{i-1} by adding a path between some two points x_i and y_i of Q_{i-1}. (This sequence is called an ear decomposition.)*

In his elegant paper, Robbins explained his result in terms of one-way streets and robustness in terms of repairs on a given street. It follows from his proof that (iii) can be extended to say that Q_0 can be chosen to contain any given point, in our case the root of Q.

Our motivation for what follows is the proof of Gal (1979) that $V^p = \mu/2$ for Eulerian networks. He simply takes any Eulerian tour of Q, equiprobably with its reverse tour, as the Searcher mixed strategy. For expanding pointwise searches $P(t)$, unlike pathwise searches, the reverse function $P(\mu - t)$ may not be a pointwise (expanding) search. In order to adapt Gal's idea to the expanding search context we need to assume that the pointwise search $P : [0, \mu] \to Q$ is *reversible*, by which we mean that $P^{-1}(t) = P(\mu - t)$ is a pointwise expanding search (see Definition 12). Note in particular that a reversible pointwise search must end at $P(\mu) = P^{-1}(0) = O$, the root.

Theorem 21 *A network Q is 2-arc connected if and only if it has a reversible combinatorial search.*

Proof. First suppose Q has a reversible combinatorial search. Suppose an arc a is traversed between times t_1 and t_2, that is, $a = P(\{t : t_1 < t < t_2\})$. Then $Q - a = P([0, t_1)) \cup P^{-1}([0, \mu - t_2))$ is the union of connected sets with a common point $O = P(0) = P^{-1}(0)$ so it must be connected. Hence Q is 2-arc connected.

Now suppose Q is 2-arc connected. Then by Theorem 21 it has an ear decomposition starting with a cycle, $Q_0, Q_1, ..., Q_k$. We can assume that the cycle, Q_0 includes O. We construct reversible combinatorial searches P_i on Q_i inductively, where the P_i are sequences of arcs, with no arcs repeated. Let S_0 be the cycle on Q_0 starting at O. This is clearly a reversible combinatorial search on Q_0. Assume we have constructed P_i, a reversible pointwise search on Q_i, for $1 \leq i < k$. We have $Q_{i+1} = Q_i \cup A_i$, where A_i is a path from a node $x \in Q_i$ to a node $y \in Q_i$. We can assume that x occurs before y in P_i, otherwise we can relabel the nodes. Let P_{i+1} be the combinatorial search on Q_{i+1} which follows P_i until reaching x, then follows the path A_i from x to y, and finally follows the remainder of P_i from x to O. Then P_{i+1}^{-1} consists of the following three expanding arc sequences: firstly the path along P_i^{-1} from O to x, next the path along A_i^{-1} from y to x, and finally the path along P^{-1} from x to O. Each of these expanding arc sequences starts from a point that has already been reached by P_{i+1}^{-1}, so P_{i+1}^{-1} is a combinatorial expanding search, and P_{i+1} is reversible, as required. ■

We can now solve the expanding search game for any 2-arc connected network.

Corollary 22 *If Q is 2-arc connected then $V = \mu/2$. An optimal Hider strategy is the uniform distribution and an optimal Searcher strategy is an equiprobable mixture σ of the pointwise searches P and P^{-1}, where P is any reversible pointwise search.*

Proof. Note that if $T(P,H) = t$ then $P(t) = H$ and hence also $P^{-1}(\mu - t) = H$. Consequently $T(P^{-1}, H) = \mu - t$, and hence

$$\frac{1}{2}T(P,H) + \frac{1}{2}T\left(P^{-1}, H\right) = \frac{t}{2} + \frac{(\mu - t)}{2} = \frac{\mu}{2}. \tag{17}$$

This shows that $V \leq \mu/2$ and the reverse inequality is given in Lemma 17. ■

It may be worth noting that in the pathwise search game context, the so called *three-arc network,* consisting of three unit length arcs between two nodes, was notoriously difficult to solve. See Pavlovic (1995). In the expanding search context however, it is easy because it is $2-$arc connected, and so has value $3/2$.

5.4 Network decomposition

In the previous two sections we have found formulae for the value of the expanding search game $\Gamma(Q)$ when Q is a tree and when it is 2-arc connected. In fact these results give bounds on the vale $V(Q)$ of an arbitrary network.

Corollary 23 *For any rooted network Q, of total length μ, we have*

$$\mu/2 \leq V(Q) \leq \mu. \tag{18}$$

The left side is an equality iff Q is $2-$arc connected; the right side holds with equality iff Q is the single arc, interval network I_μ with the root at one end.

Proof. The left inequality is Lemma 17. The right side is obvious because in fact every expanding search has maximum search time μ. If Q is not $2-$arc connected it has a disconnecting arc a, and we may modify the uniform measure u to u_a by moving all the measure of a onto its end away from the root. The same arguments used in the proof of Lemma 17 show that for any search S, $T(S, u_a) \geq \mu/2 + u(a) \cdot \lambda(a)/2$. That is, in the case the Hider was originally on a, it will be found at a time $\lambda(a)/2$ later, on average. If Q is a tree other than I_μ, then $D < \mu$, so Theorem 18 implies $V(Q) < \mu$. If Q is not a tree, it has a cycle, and as in the pathwise search theory we consider the Eulerian network Q' in which all arcs not in the cycle are doubled. The Eulerian tour S of Q' starting at the root is a reversible pointwise expanding search of Q of length $\mu' = \mu(Q') < 2\mu$, so the inequality (17) holds with μ' replacing μ, so $V(Q) \leq \mu'/2 < \mu$. ■

Since $V \leq V^p$, it is not surprising that, although the bounds in (18) are the same as those for the pathwise search value, the set of networks where it holds with equality are larger in expanding search for the lower bound (all 2-arc connected networks; not just Eulerian networks) and smaller for the upper bound (only the interval network; not for all trees).

6 Conclusions

This paper has introduced a new type of network search, in which the searched area expands continuously at a unit rate. We show how to optimize such searches to minimize the expected time to find a Hider of known distribution and to minimize the worst case expected search time for a Hider of unknown distribution. The latter essentially extends the theory of network search games to the context of expanding search.

A similar expanding search approach could be taken to the problem of rendezvous search (see Anderson and Essegaier (1995), Lim (1997), Gal (1999), Alpern (2002), Gal and Howard (2005), Chester and Tütüncü (2004)). Two rendezvousers (each perhaps representing a tribe or animal colony) expand their searches until one of them reaches a point previously searched by the other. We may assume that information (tracks or scent markers, for animals or ant colonies) has been left in the original search which solves the meeting problem. This is only one of many possible extensions of the theory. Other extensions include (i) Searcher picks the starting point of the search, (ii) time to traverse an arc depends on the direction, (iii) there is a cost involved in searching a node.

Another interesting area for further research has been suggested by an anonymous referee. Our approach, even to the Bayesian problem, does not consider algorithms or complexity. For example on tree we simply say for the Searcher to 'start on the arc leading to the subtree of maximal hider density', but we don't provide any algorithm for determine such a subtree. Perhaps algorithms analogous to path-planning could be of use in this respect. An alternative game suggested by this referee involves the Hider choosing an arc and the Searcher applying group-testing (see for example Du and Hwang (2000)) using vertex sets to recursively find him.

S. Alpern (2002). Rendezvous search: A personal perspective. *Operations Research* 50, no. 3, 772-795.

S. Alpern (2010). Search games on trees with asymmetric travel times. *SIAM J. Control Optim.* 48, no. 8, 5547-5563.

S. Alpern (2011a). A new approach to Gal's theory of search games for immobile hiders on a network. *Dynamic Games and Applications*, in press.

S. Alpern (2011b). Find-and-fetch on a tree. *Operations Research* 59, no. 5, 1246-1257.

S. Alpern and S. Gal (1988), A Mixed Strategy Minimax Theorem without Compactness, *SIAM J. Control and Optim.* 26, 1357-1361.

S. Alpern and S. Gal (2003). The Theory of Search Games and Rendezvous. *Kluwer International Series in Operations Research and Management Sciences*, 319 pp, Kluwer, Boston,.

S. Alpern and J. V. Howard (2000). Alternating search at two locations. *Dynamics and Control* 10, no. 4, 319-339.

S. Alpern and T. R. Lidbetter (2010). Searching a variable speed network. London School of Economics, Management Science Working Paper Series, LSEOR 10.129.

E. J. Anderson and M. Aramendia (1990). The search game on a network with immobile hider. *Networks* 20, 817-844.

E. J. Anderson and S. Essegaier (1995). Rendezvous search on the line with indistinguishable players. *SIAM J. Control Optim.* 33, 1637-1642.

V. J. Baston and F. A. Bostock (1990). An evasion game on a finite tree. *SIAM J. Control Optim.* 28, no. 3, 671-677.

E. J. Chester and R. Tütüncü (2004). Rendezvous search on the labeled line. *Operations Research* 52, 330-334.

R. Dechter and J. Pearl (1985). Generalized best-first search strategies and the optimality of A*. Journal of the ACM 32, 505-536.

D. Du, F. Hwang (2000), Combinatorial group testing and its applications. World Scientific Pub., Singapore.

S. Gal (1979). Search games with mobile and immobile hider. *SIAM J. Control Optim.* 17, 99-122.

S. Gal (1999). Rendezvous search on the line. *Oper. Res.* 47, 974-976.

S. Gal (2000). On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory* 59, 533-542.

S. Gal (2011). Search games. *Wiley Encylopedia of OR and MS.* Wiley.

S. Gal and J. V. Howard (2005). Rendezvous-evasion search in two boxes. *Operations Research* 53, 689-697.

A. Garnaev (2000). Search Games and Other Applications of Game Theory *(Lecture Notes in Economics and Mathematical Systems* Vol. 485), Springer.

J.C. Gittins (1989). Multi-armed bandit allocation indices. Wiley, Chinchester and New York.

J. C. Gittins and D. M. Roberts (1979). The search for an intelligent evader concealed in one of an arbitrary number of regions. *Naval Res. Logis. Quart.* 26, no. 4, 651-666

R. Isaacs (1965). Differential Games, Wiley, New York.

A. Jotshi and R. Batta (2008). Search for an immobile entity on a network. *European Journal of Operational Research* 191, no. 2, 347-359

K. Kikuta (1995). A search game with travelling cost on a tree, *J. Oper. Res. Soc. Japan*, 38, no. 1, 70-88.

K. Kikuta and W. Ruckle (1994). Initial point search on weighted trees. *Naval Res. Logistics* 41, no. 6, 821-831.

S. LaValle (2006). Planning Algorithms. *Cambridge University Press.*

W. S. Lim (1997). A rendezvous-evasion game on discrete locations with joint randomization. *Adv. Appl. Probab.* 29, 1004-1017.

N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou (1988). The complexity of searching a graph. *J. ACM* 35, 18-44.

L. Pavlovic (1995). A search game on the union of graphs with immobile hider. *Naval Res. Logistics* 42, no. 8, 1177-1199.

J. H. Reijnierse and J. A. M. Potter (1993). Search games with immobile hider. *Int. J. Game Theory* 21, 385-394.

D. J. Reyniers (1995). Coordinating two searchers for an object hidden on an interval. *Journal of the Operational Research Society* 46, 1386-1392.

D. J. Reyniers (1996). Coordinated search for an object hidden on the line. *European Journal of Operational Research*, 95, 663-670.

D. M. Roberts and J. C. Gittins (1978). The search for an intelligent evader: strategies for searcher and evader in the two-region problem. *Naval Res. Logis. Quart.* 25, no. 1, 95-106

L. C. Thomas (1992). Finding your kids when they are lost. *Journal of the Operational Research Society* 43, 637-639.