TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2013

 | ISBN 978-0-9843378-4-2
http://dx.doi.org/10.1287/educ.2013.0117

Network Search from a Game Theoretic Perspective

Steve Alpern
Operational Research and Management Sciences Group, Warwick Business School, University of Warwick, Coventry CV4 7AL, United Kingdom, steve.alpern@wbs.ac.uk

Abstract A searcher wishes to find an unknown point H on a given metric network Q. Depending on what the network represents, H might be a file, an explosive device, or an adversary. The searcher typically starts from a known point O and moves at unit speed on Q until the first time T that he reaches H. The approach studied here, formulated by Isaacs [21] (R. Isaacs. *Differential Games.* John Wiley & Sons, New York, 1965) in his 1965 book *Differential Games*, is to view this problem as a zero-sum game with the capture time T as payoff. The hider (who chooses H) is the maximizer, and the searcher is the minimizer. The approach is equivalent to viewing the searcher's problem as a game against nature, but it also gives any real hider advice as to the optimal distribution of choosing H. The "classical theory" begins in 1979 with the work of Gal [16, 18] (S. Gal. Search games with mobile and immobile hider. *SIAM Journal on Control and Optimization* 17:99–122, 1979; S. Gal. On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory* 29:533–542, 2000) for trees Q, and culminates with Gal's extension of the tree theory to weakly Eulerian networks in 2000. In between, many other researchers contributed to the theory. Here, we review the classical theory and present extensions in various directions that enlarge the applicability and theory of such search games.

Keywords search game; game theory

1. Introduction

A searcher, starting at a given point in a known network, wishes to reach (and thus "find") an object hidden at an unknown point in the network, not necessarily at a vertex. The edges of the network have given lengths, and the searcher can move along these at unit speed. The question dealt with in this tutorial is how the searcher should move along the network so as to minimize the expected time to find the object, called the capture time T. It turns out that a randomized, or mixed, strategy of choosing his path is usually required. The approach taken is that of a game against nature, where we assume that the object is hidden by an adversarial hider (or nature), who wants to maximize the time to find it. In essence, we are analyzing the game of *hide-and-seek* on a network. Games of this type, including variations where the hidden object is a mobile agent, go under the name of *search games.*

The subject of search games was first formalized in the final chapter of Isaac's [21] monograph *Differential Games.* That book was mainly concerned with differential games of perfect information, where each player was informed of all previous moves of the other. Typical games covered there were pursuit–evasion games, where the pursuer wished to minimize the time taken to reach a visible evader, who in turn could see the approach of the pursuer. In the final, forward-looking chapter, Isaacs [21] asked what would happen to games like pursuit–evasion if both players lost all information about the previous actions (and hence current location) of the other player. In other words, the scene of the action would be plunged into darkness, turning pursuit–evasion games into search games.

This tutorial will concentrate on search games where the search region is a network and the hider is immobile. We will formally define these games in §2. The main class of networks for which these games have been solved are the so-called weakly Eulerian networks. Roughly speaking, these are networks consisting of disjoint Eulerian networks connected in a treelike fashion. This class includes as special cases Eulerian networks and trees. Eulerian networks are easy to solve, and the ground breaking paper of Gal [16] gave a solution for all trees, which we present in §3. Section 4 presents the extension to weakly Eulerian networks given by Gal [18], following an intermediate extension to the smaller class of weakly cyclic networks given by Reijnierse and Potter [27]. Outside the class of weakly Eulerian networks, little is known about the solution of search games. A notable exception is the work of Pavlovic [26], who solved the game for the network consisting of three unit length edges connecting two vertices, one of which is the searcher starting point. This work is described in §5. These sections constitute a discussion of what we consider the "classical theory" of search games, roughly what was known at the time of the monograph of Alpern and Gal [5].

The remaining sections cover more recent developments in which some of the basic assumptions of the classical theory (as covered in §2) are relaxed. Section 6 relaxes the assumption that the searcher is constrained to start at a given point, known to the hider. It describes the work of Dagan and Gal [15], Alpern [1], and Alpern et al. [7, 8], which extend some but not all of the fixed start theory to arbitrary start. Section 7 relaxes the assumption that each edge can be traversed in the same time in either direction. It turns out that relaxing this assumption allows the network search paradigm to encompass search problems that were previously thought outside of its range. This section covers search games on variable speed networks introduced by Alpern [2] and further analyzed by Alpern and Lidbetter. Section 8 removes the restriction that the searcher follows a continuous path in the network, allowing a more general increasing nested family of searched sets $X(t)$ that start at root vertex and expand to encompass the whole network Q. We call this paradigm *expanding search*. Section 8 explores both the Bayesian problem of how to search for a hider with a known distribution and the search game on trees and other networks. An outgrowth of this investigation for the star graph is the solution of the multiple hider problem recently given by Lidbetter [25].

Other significant contributions to the theory of network search include Anderson and Aramendia [11], Baston and Bostock [12], Baston and Kikuta [13], Jotshi and Batta [22], and Kikuta and Ruckle [24].

2. The Search Game $\Gamma(Q,O)$

The network Q on which the search game Γ is played out is is best described as a metric space (Q,d). In this setting, the pure strategies available to the searcher are the unit speed paths covering Q, which start at the point O. We adopt the notational custom of using uppercase letters to denote pure strategies and lowercase letters to denote mixed strategies. Thus the pure strategy set for the searcher is given by

$$\mathcal{S} = \{S\colon R^+ \to Q,\ \ S(0) = O,\ \ d(S(t), S(t')) \le |t - t'|\}.$$

The pure strategy set $\mathcal{H}$ of the hider is simply all points in Q, that is $\mathcal{H} = Q$. The payoff is given by the capture time T, defined as

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

Observe that if $T(H,S) > t_1$ (capture has not occurred by time t_1), then there is a minimum distance (positive) from the search path so far to H, so if H and S are perturbed small enough (using uniform topology for S), they will still not meet by time t_1. In other words, T is lower semicontinuous. Since $\mathcal{S}$ is compact in the uniform topology, this is sufficient to ensure that the game has a value $V = V(Q,O)$. This was proved directly by Gal [17] and can now be seen to follow from a more general minimax theorem of Alpern and Gal [5].

This result established moreover the existence of an optimal mixed strategy for the searcher and ε-optimal strategies for the hider. We adopt the same notation T, applied to mixed strategies, to denote the *expected* capture time. Thus $T(h,s)$ is the expected capture time when the hider adopts the mixed strategy h and the searcher adopts the mixed strategy s.

There are some elementary bounds on $V(Q,O)$ that we may obtain using two mixed strategies: one that is always available to the hider and one that is always available to the searcher.

If the hider adopts the uniform hiding distribution h_u (hiding on each edge with probability proportional to its length and uniformly on each edge), then it follows from the unit speed restriction on the searcher that in any time period of length $\triangle$ he can find the hider with probability at most $\triangle/\mu$, where $\mu = \mu(Q)$ is the total length of Q. Consequently, the expected capture time if the hider adopts h_u is $\mu/2$, or

$$V \geq \mu/2. \tag{1}$$

For the searcher, we consider a tour $S(t)$ of Q with minimum length $\bar{\mu}$, a so-called Chinese postman tour. Suppose he equiprobably adopts the tour S and the reverse tour $S'(t) = S(\bar{\mu}-t)$, a *random Chinese postman tour* (*RCPT*) $\bar{s}$. If $T(H,S) = t_0$, then it follows that $S'(\bar{\mu}-t_0) = S(t_0) = H$, so $T(H,S') \leq \bar{\mu} - t_0$. Thus

$$T(H,\bar{s}) \leq \frac{1}{2}t_0 + \frac{1}{2}(\bar{\mu}-t_0) = \frac{\bar{\mu}}{2}.$$

Since the mixed strategy $\bar{s}$ is always available to the searcher, we have

$$V \leq \frac{\bar{\mu}}{2}. \tag{2}$$

Combining the two estimates on the value, we have, for all networks (Q,O),

$$\frac{\mu}{2} \leq V(Q,O) \leq \frac{\bar{\mu}}{2}. \tag{3}$$

For Eulerian networks we have that $\mu = \bar{\mu}$ and hence $V = \mu/2$. The next two sections trace the identification of those networks for which the upper bound is tight, that is, the characterization of simply searchable networks.

Definition 1. A rooted network (Q,O) is called *simply searchable* if $V(Q,O) = \bar{\mu}/2$.

In the next two sections we will show that trees are simply searchable and then that weakly Eulerian networks are identical with simply searchable networks. (In both cases the value does not depend on the location of the root O.)

3. Search Games on Trees

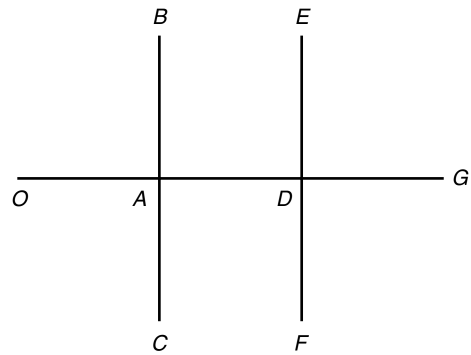
For rooted trees, it is clear that any hiding point H that is not a leaf vertex is dominated by any leaf vertex that it disconnects from the root, so we may assume the hider is at a leaf of the tree. For rooted trees, it is easily seen that a tour from the root must traverse each edge twice, so $\bar{\mu} = 2\mu$. So for a tree to be simply searchable, its value must be equal to its total length, $V = \mu$. This result was established in Gal [16]. Here we outline an alternative approach to this result taken from Alpern [4] that gives an alternative to the searcher to the RCPT (the random Chinese postman tour defined in Gal [16] as an equiprobable mixture of a Chinese postman tour and its time-reversed tour) optimal mixed strategy.

The optimal mixed strategy for the hider is unique and can be described as follows.

Definition 2. If (Q,O) is a rooted tree, every edge a determines a unique branch $\sigma(a)$ consisting of a and all edges connected to the root through a. *The equal branch density* (*EBD*) *hider distribution* h_e is the unique distribution concentrated on the leaves of Q that assigns probabilities to the branches at every vertex proportional to their total lengths (or equivalently, to the time required to tour them).

To illustrate the calculation of h_e, consider the rooted tree drawn in Figure 1, where all edges have length 1. At the first branch vertex A, the three branches have lengths 1, 1,

Figure 1. A rooted tree.



and 4, and so get respective probabilities 1/6, 1/6, and 4/6. Thus $h_e(B) = h_e(C) = 1/6$; that is, the hider chooses B and C as H with probability 1/6 each. Next, the three branches at D equally divide the remaining probability of 4/6, so they get $4/(6 \cdot 3) = 2/9$ each. So $h_e(E) = h_e(F) = h_e(G) = 2/9$. The recursive definition of EBD given here comes from Gal [16].

When searching for an object hidden according to the EBD distribution, the optimal pure response strategies are the *depth-first* (DF) paths, as shown more generally in Alpern and Lidbetter [10]. These are search paths that leave a vertex by an untraversed forward (away from root) edge whenever this is possible. (If all such edges have been traversed already, they go back on the unique edge toward the root.) For example, the path OABADEDACADFDG is not DF because from D it chooses edge DA when DF has not yet been traversed. We note that Chinese postman tours on trees are DF paths (which return to the root after covering the tree).

A particular mixture of DF paths is obtained from the following.

Definition 3. *The random depth first* (*RDF*) *searcher strategy* **r** is the mixed searcher strategy determined by the behavioral rule of choosing equiprobably among the untraversed forward edges at a vertex.

It can be shown (Alpern [2]) that the expected time for a searcher following r to reach any leaf L of the tree is exactly μ (total length of Q). It follows that any leaf is an optimal hider response to a searcher using r and hence also any mixture of these such as the EBD strategy h_e. Similarly, since any DF strategy is an optimal response to h_e, so is any mixture such as r. Thus we have the following (Alpern [2]):

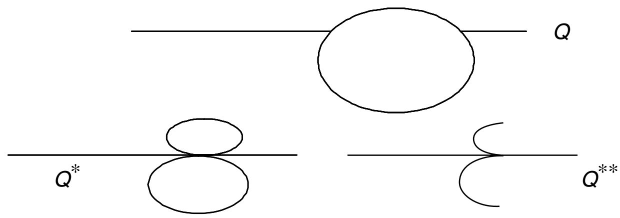
Theorem 4. *The EBD strategy h_e for the hider and the RDF strategy r for the searcher are optimal responses to each other, and thus (h_e, r) is a Nash equilibrium.*

Since Γ is a zero-sum game and $T(h_e, r) = \mu$, this implies Gal's [16] theorem that $V(Q, O)$ is μ for any rooted tree, or that *trees are simply searchable*. Thus the RCPT is another optimal mixed strategy for the searcher. We know that all optimal mixed searcher strategies are concentrated on the DF paths. The RCPT and RDF are extreme mixtures in that the former uses only two DF paths and the latter uses them all. There are many in between mixtures that are optimal. It is shown in Alpern and Lidbetter [9] that they all have a certain property that as behavioural strategies they are unique: *When reaching any branch vertex of the tree for the first time, the two branches* (*assuming a binary tree*) *are searched first with equal probability.*

4. Search Games on Weakly Eulerian Networks

Thus far we have seen that both Eulerian networks and trees are simply searchable (have $V = \bar{\mu}/2$). What other classes of networks have this property? Reijnierse and Potter [27]

FIGURE 2. Identification–cutting–deleting technique.



showed that if there are at most two disjoint paths between any two vertices of Q (that is, if Q is weakly cyclic), then Q is simply searchable. Subsequently, Gal [18] extended that result to the following wider class of networks. Roughly speaking, these are networks containing a family of disjoint Eulerian subnetworks, connected in a treelike fashion. A more formal definition is given below.

Definition 5. A network Q is called *weakly Eulerian* if the removal of all (open) edges which disconnect it leaves an Eulerian network.

Clearly, this class includes all trees (the Eulerian network here is just a union of vertices, as they all have even degree 0) and Eulerian networks. The following elementary result is useful in proving results concerning the search value $V(Q)$ of a network Q.

Lemma 6 (Edge Adding). *Let the network Q' be obtained from a network Q by adding an edge a of length $l \geq 0$ between two points x and y of Q. The root can be any point in Q. Then,*

1. *$V(Q') \leq V(Q) + 2l$, so that $V(Q') \leq V(Q)$ if we simply identify the vertices ($l = 0$). Note that the first part applies even if a is attached only at one of its ends.*
2. *If $l \geq d_Q(x, y)$, then $V(Q') \geq V(Q)$. In particular, any hiding strategy on Q' does equally well (in the worst case) on Q.*

To illustrate the use of this lemma, consider a weakly Eulerian network Q with a single Eulerian component, as drawn in Figure 2. We demonstrate a technique we call *identification–cutting–deleting.* Let Q^* be the network obtained by identifying to a single vertex all vertices of Q that lie on the Eulerian component. By part (1) of the lemma, we have $V(Q^*) \leq V(Q)$. The network Q^* has a number of loops at the identified vertex. Cut each of these in half, removing one of the halves, to obtain a new network Q^{**}. We may think of Q^* as being obtained from Q^{**} by adding an identical edge to each leaf edge at the identified vertex, converting it to a loop of twice the length. Then (viewing Q^* as Q and Q^{**} as Q' in part 2 of the Lemma) we see that $V(Q^*) \geq V(Q^{**})$, and consequently $V(Q^{**}) \leq V(Q)$. But Q^{**} is a tree with the same length μ as the weakly Eulerian network Q, so by the results of the previous section, we have $V(Q) \geq \bar{\mu}/2 = \mu$, and by (2) we have $V(Q) = \bar{\mu}/2$; that is, Q is simply searchable. Moreover, Gal [16] proved the following topological characterization of simply searchable networks.

Theorem 7. *A network Q is simply searchable if and only if it is weakly Eulerian. For such networks, the location of the root does not affect the value.*

5. The Three-Edge Network

Aside from weakly Eulerian networks, few networks have been solved. A notable exception is what we will call the "three-edge network" Q_3, consisting of two vertices O and A connected by three identical unit length edges. For this network we have $\mu = 3$ and $\bar{\mu} = 4$, so by the general bounds on the value (3) we know that $1.5 \leq V(Q_3, O) \leq 2$. The following randomized search strategy, originally suggested by D. J. Newman, turns out to be optimal: Go from O to A on a random edge (called the first edge). Go from A back toward O on one of the remaining edges (called the second edge) a random distance α to a point A_a before reversing back to A. Then go on the last edge to O and then back on the second edge to A_α. The random distance α should be chosen according to the cumulative probability distribution $1/2 + e^\alpha/4$ on the interval $[0, \ln 2]$. This guarantees an expected search time of no more than $(4 + \ln 2)/3 \approx 1.5644$. The optimal hiding strategy is to hide equiprobably on the edges at a distance β from A, where β has cumulative probability distribution $2e^{-\beta}$ for β in $[0, \ln 2]$. Summarizing, we have the result of Pavlovic [26].

Theorem 8. *The value of the three edge game is given by*

$$V(Q_3, O) = (4 + \ln 2)/3.$$

An earlier discrete analysis of this problem was given by Bostock [14].

6. Arbitrary Searcher Starting Point

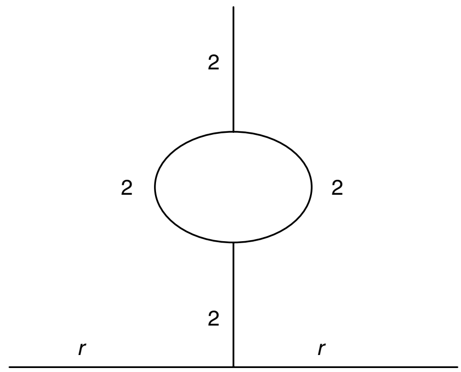
In the field of search games it is generally assumed that the searcher must start at a given point of the network, which is known to the hider. However, there is a small body of work that makes the contrary assumption, that the searcher can start anywhere (and thus his starting point is unknown to the hider). In this context the role earlier played by weakly Eulerian networks is now taken up, to some extent, by the smaller class of semi-Eulerian networks. We define a semi-Eulerian network as a tree to which disjoint Eulerian networks are attached at single points. The role taken by Chinese postman tours (of length $\bar{\mu}$) is replaced by Chinese postman paths (minimum length covering paths) of length we denote by $\hat{\mu}$. A random Chinese postman path is simply an equiprobable mixture of a Chinese postman path (from end e_1 to end e_2) and its reverse path (from end e_2 to end e_1). Clearly, such a strategy reaches every point in Q in time not exceeding $\hat{\mu}/2$, so this is an upper bound on the value $\hat{V} = \hat{V}(Q)$ for any network. Here, $\hat{V}$ stands for the value of the arbitrary start game. For arbitrary start games, we say that Q is simply searchable if $\hat{V}(Q) = \hat{\mu}/2$. Dagan and Gal [15] initiated the study of arbitrary start search games, proving that trees are simply searchable, a result extended by Alpern [1] to the following.

Theorem 9. *If Q is semi-Eulerian, then it is simply searchable in the arbitrary start sense.*

This result is generalized in Alpern et al. [7] to show that we cannot only attach Eulerian networks to a tree, but they can be weakly Eulerian as long as they do not increase the diameter of the tree. This paper also demonstrates that, unlike Gal's [18] result (Theorem 7) for fixed start networks, there is no topological characterization of simply searchable networks for the arbitrary start case. Consider the family of networks Q_r drawn in Figure 3, consisting of four edges of fixed length 2 and two horizontal edges at the bottom, each of variable positive length r. If r is sufficiently large, the diameter is $2r$, and this will not change if we add or delete the vertical part. So by the observation above about adding a weakly Eulerian network of small diameter, this network will be simply searchable: An optimal search strategy is to start at the left or right bottom; when reaching the vertex at the bottom of degree 3, search the vertical part in time 12, and then continue to the other lower end. However, if it can be shown that the simple network with $r = 0$ is not simply searchable, it follows that for sufficiently small r it is also not simply searchable. All of these networks for $r > 0$ are topologically the same, but the answer to the simply searchable question depends on r.

The analysis of arbitrary start games on symmetric networks is given in Alpern et al. [8].

FIGURE 3. A family of networks Q_r, $r > 0$.



7. Asymmetric Travel Times

The classical theory assumes that an edge can be searched in the same time from either end, that is, regardless of the direction of search. It turns out that allowing distinct travel times in the two directions increases the applicability of search games. We will see this enhanced applicability in the subsequent sections, but for the moment it is useful to simply take the view that the network is drawn on a nonlevel surface and that some directions are uphill (slower, longer travel times) and others are downhill (faster, shorter travel times).

Asymmetric travel times were incorporated into the theory of search games by Alpern [2], who gave each edge a a forward and reverse direction and specified two travel times, F_a and R_a.

For trees, the direction away from the root was always taken as forward. It turns out that for asymmetric (travel time) trees, the random Chinese postman tours are no longer optimal search strategies. However the random depth-first strategy can be modified to an optimal *branching* strategy. For simplicity we restrict our discussion to binary trees (two branches at every branching vertex). A branching function is a function q that assigns to each branching (nonleaf) vertex a probability distribution over the branching edges (the edges at that vertex going away from the root). Associated with q is a branching search strategy $B(q)$ defined as follows.

1. When leaving a branch vertex for the first time, choose the leaving edge a with probability $q(a)$.
2. When leaving a branch vertex for the second time, choose the forward edge not taken earlier.
3. When leaving a branch vertex for the third time, take the unique edge leading back toward the root.

Note that the random depth-first search described in §3 is the branching function where q is the equiprobable distribution $(1/2, 1/2)$ at every branch vertex. It is optimal for symmetric networks where $F_a = R_a$ for every edge a. For asymmetric networks, the calculation of the optimal branching function is more complicated—a recursive method is given for trees (5) in Alpern [2].

For the hider, the EBD distribution is still optimal, if one uses the parenthetical phrase in the definition: The probabilities assigned to two branches at a common vertex are proportional to the time taken to tour them. The time taken to tour a branch of a variable speed tree (one with asymmetric travel times) is the sum of all forward and reverse times on its edges.

For asymmetric trees, a very simple formula for the search value was found by Alpern and Lidbetter [10]. It uses the intuitive notion of the *height* $\delta(x)$ of any point x on Q as the difference in time required to reach x from 0 and the return time from x to O. The *incline* $\triangle$ of Q is the mean height of its leaf vertices, with respect to the EBD distribution. If Q is symmetric, then all heights are zero and so is the incline. The tour time τ of Q is the time required to tour Q from the root. It is simply the sum of $F_a + R_a$ over all edges a. If Q is symmetric, then $\tau = 2\mu$, twice the total length of the network.

Theorem 10. *If Q is a tree with asymmetric travel times, the search value $V = V(Q,O)$ is given by*

$$V = \frac{1}{2}(\tau + \Delta). \tag{4}$$

Note that this reduces to the known result $V = \mu$ for symmetric travel time trees.

We can also use the notion of the incline to give a simple formula for the optimal branching function. For any branching vertex x, denote by a and b the forward edges at x, and by $\sigma(a)$ and $\sigma(b)$ the subtrees rooted at x following a and b, respectively. Let $\triangle_a$ and $\triangle_b$ denote the inclines of the rooted trees $\sigma(a)$ and $\sigma(b)$, and let $\tau(x)$ denote the tour time of the subtree at x, that is, $\sigma(a) \cup \sigma(b)$. Then when leaving vertex x for the first time, the searcher should take edge a with probability

$$q(a) = \frac{1}{2} + \frac{1}{2\tau_x}(\Delta_a - \Delta_b); \tag{5}$$

that is, the bias away from equiprobability is toward the branch with a higher incline—the searcher is biased toward going up. For symmetric trees, both inclines $\triangle$ are zero, so the searcher branching is unbiased.

For asymmetric networks, things can become complicated. Even for a network that is topologically the same as a circle, the optimal solutions can involve backtracking, as in the following example taken from Alpern and Lidbetter [10].

Theorem 11. *Consider the network $U(b)$ consisting of two identical edges from the root O to another vertex A with forward travel time 1 and reverse time $b \geq 1$. Then,*

1. *The search value is $V = V(U(b),O) = 1 + \frac{1}{2}(b-1)\ln 2$.*
2. *An optimal strategy for the hider is to pick x according to the density function $4e^{-2x}$ on the interval $[0, \ln 2/2]$. Then he hides equiprobably on the two points at forward distance x from O.*
3. *An optimal strategy for the searcher begins by choosing a number y from $[0, \ln 2/2]$ according to the density function $2e^{2y}$. With probability $p = (b+3)/(2b+2)$, he tours the circle equiprobably in either direction. With probability $1-p$, he goes around in an equiprobable direction until he is at forward distance y from O, then reverses direction and goes around the circle until he has reached all points.*

In §3 we used a technique we call *identification–cutting–deleting* to give a lower bound on the value for weakly Eulerian symmetric networks. The same technique can be used on arbitrary asymmetric networks to obtain the following lower bound on the network (Alpern and Lidbetter [10]).

Theorem 12. *Let Q, O be a rooted network with every edge a oriented so that $F_a \geq R_a$. Let $\omega = \sum_{a\in\mathcal{A}} R_a$ denote the time required to traverse all the edges of Q in the quicker direction. Then, with ρ_a denoting the in-radius of a,*

$$V(Q) \geq \frac{1}{2}\left(\omega + \sum_{a\in\mathcal{A}} \frac{R_a}{\omega}(2\ \rho_a - R_a)\right). \tag{6}$$

7.1. Application to Games with Search and Travel Costs

Kikuta [23] investigated a search game $K = K(Q,O)$ on a time-symmetric rooted tree Q, O where each vertex i is assigned a search cost $c_i \geq 0$, with $\sum c_i = C$. When encountering a vertex, the searcher can either search it at cost (time loss) c_i or bypass it (to search it later) without incurring a cost. It turns out that the game K can be represented by a traditional (travel cost only) search game on an asymmetric network Q'. The network Q' is obtained by replacing the search costs with *search edges* a_i between each vertex i of Q and a new leaf vertex i' of Q', with $F_{a_i} = c_i$ and $R_{a_i} = 0$, so that $D_{a_i} = c_i$. By applying the value formula (4) to Q', an associated formula for Kikuta's game was obtained in Alpern and Lidbetter [10].

Theorem 13. *The value of Kikuta's game K on a rooted time-symmetric tree Q, O of total length μ and search costs c_i totalling to C is given by*

$$V = \mu + \frac{1}{2}\left(C + \sum_{\textit{vertices } i \textit{ of } Q} e(i') \cdot c_i\right), \tag{7}$$

where e is the EBD distribution on the associated Q'. If the costs at all n vertices of Q are equal to c, then $V = (1/2)(\tau + (n+1)c)$ and the random Chinese postman tour (and searching every vertex when you come to it) is optimal.

7.2. Application to Find-and-Fetch Problems

A problem related to asymmetric networks is that of find-and-fetch. Here, the searcher has to find the hider and bring him back to the starting point. The return speed (carrying the hider) can be different from the unit speed of search. If one is modeling search and rescue, or a predator bringing large prey back to the nest, the return speed is lower than the search speed. However, if the lost object is a contact lens, the search speed is slow (looking for a small object), although return speed can be fast. So there are some connections with asymmetric travel times. However, in a general network, one does not know at the outset which is the fast direction on an edge. A precursor of the general value formula (4) is given in Alpern [3].

Theorem 14. *The value (optimal find-and-fetch time) of the game on a rooted tree of total length μ, with unit search speed and return speed r, is given by*

$$V = \mu + D/r,$$

where D is the mean distance from the root to the leaf vertices, with respect to the EBD hider distribution.

8. Expanding Search

In all of the search games we have discussed so far, the searcher follows a continuous path. Another model of search, proposed in Alpern and Lidbetter [9], is where the searcher chooses a nested family of subsets $X(t) \subset Q$, with $X(0) = O$ (the root), $X(t) \subset X(t')$ for $t < t'$, and $\lambda(X(t)) = t$, where $\lambda(A)$ denotes the total length of a subset $A \subset Q$. The interpretation is that $X(t)$ represents the portion of Q that has been searched by time t. We call this search paradigm *expanding search*, as the region searched expands at unit rate from just the start point O to the whole of the search region ($X(\mu) = Q$). In some cases, it may be expanding only in one direction at a time, so we can talk of the location of the searcher, but in other cases we would need to describe the search using a group of searchers.

Unlike traditional continuous search, for expanding search we can solve the Bayesian problem on trees; that is, if there is a known hider distribution on a rooted tree, we can determine the optimal expanding search. In such a case we need only consider sequences of edges such that the first edge touches the root and every successive edge touches a previously chosen edge. The Bayesian solution involves the notion of the search density of a subtree, the ratio of the probability the hider is in that subtree to the total length of the subtree. Assuming all subtrees have distinct search densities (a nondegeneracy assumption), the following is shown in Alpern and Lidbetter [9].

Theorem 15. *Given a known hider distribution on the leaves of a rooted tree, the optimal expanding search begins by searching the subtree of maximum search density.*

This result is sufficient to generate the entire optimal expanding search, because it always gives the optimal initial edge at the root. After the first edge is chosen, the remaining problem is the same type as the original one (on some other tree), so the theorem above can be used repeatedly.

For trees, a solution to the expanding search game is given as follows (Alpern and Lidbetter [9]).

Theorem 16. *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 $V=(\mu+D)/2$, where D is the mean distance from the root to the leaf vertices, with respect to the EBD hider distribution. The optimal hider distribution is the EBD distribution. The optimal searcher strategy is a branching strategy where at a vertex x with forward edges a and b, edge a is chosen with probability*

$$q(a)=\frac{1}{2}+\frac{1}{2\mu_x}(D_a-D_b),$$

where μ_x is the total length of the subtree rooted at x, and $D_a=D(\sigma(a),x)$, where $\sigma(a)$ is the subtree starting with edge a.

For general networks, we can obtain an analog of (3) for expanding search.

Theorem 17. *For any rooted network Q of total length μ, we have*

$$\mu/2\le V(Q)\le\mu. \tag{8}$$

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

8.1. Multiple Hiders

For the expanding search game on a star network, Lidbetter [25] found an elegant solution for the more general case where the payoff is the time required to find all of k hiders (hidden by a common adversary). This is equivalent to the problem of finding k hiders in n boxes, where there is a cost c_i to search the ith box. It is optimal for the searcher to begin by searching a k-subset G of boxes with probability proportional to the product of the search costs of the boxes in G. The searcher should then search the $n-k$ remaining boxes in a random order.

9. Conclusions

Since its formulation by Isaacs [21] in 1965, the field of search games has expanded greatly in both theory and applications. The main surveys and monographs on the subject, in chronological order, are those of Gal [17], Garnaev [20], Alpern and Gal [6], and Gal [19]. These cover a wider domain than this tutorial, including higher-dimensional search regions and mobile hiders. The material in §§7 and 8 is newer than any of those monographs. Because of restrictions on length, we have concentrated on the theory for networks and have left out many related theories (particularly mobile hiders and ambush strategies) and also recent applications to predator search for prey and antiterrorism.

We note that even the seemingly elementary game of hide-and-seek is still unsolved for most networks. New ideas regarding network topologies, computational techniques, complexity, and areas of application are greatly needed.

Acknowledgments

The author wishes to acknowledge assistance from NATO Collaborative Research [Grant CLG 983583].

References

[1] S. Alpern. Hide-and-seek games on a tree to which Eulerian networks are attached. *Networks* 52(3):162–166, 2008.

[2] S. Alpern. Search games on trees with asymmetric travel times. *SIAM Journal on Control and Optimization* 48(8):5547–5563, 2010.

[3] S. Alpern. Find-and-fetch search on a tree. *Operations Research* 59(5):1258–1268, 2011.

[4] S. Alpern. A new approach to Gal's theory of search games on weakly Eulerian networks. *Dynamic Games and Applications* 1:209–219, 2011.

[5] S. Alpern and S. Gal. A mixed strategy minimax theorem without compactness. *SIAM Journal on Control and Optimization* 26:1357–1361, 1988.

[6] S. Alpern and S. Gal. The theory of search games and rendezvous. *Kluwer International Series in Operations Research and Management Sciences*, Vol. 55. Kluwer Academic Publishers, Boston, 2003.

[7] S. Alpern, V. Baston, and S. Gal. Network search games with immobile hider, without a designated searcher starting point. *International Journal of Game Theory* 37:281–302, 2008.

[8] S. Alpern, V. Baston, and S. Gal. Searching symmetric networks with utilitarian-postman paths. *Networks* 53:392–402, 2009.

[9] S. Alpern and T. R. Lidbetter. Mining coal and finding terrorists: The expanding search paradigm. *Operations Research* 61(2):265–279, 2013.

[10] S. Alpern and T. R. Lidbetter. Searching a variable speed network. *Mathematics of Operations Research*. Forthcoming.

[11] E. J. Anderson and M. Aramendia. The search game on a network with immobile hider. *Networks* 20:817–844, 1990.

[12] V. J. Baston and F. A. Bostock. An evasion game on a finite tree. *SIAM Journal on Control and Optimization* 28(3):671–677, 1990.

[13] V. J. Baston and K. Kikuta. Search games on networks with travelling and search costs and with arbitrary searcher starting points. *Networks*. Forthcoming, 2013.

[14] F. A. Bostock. On a discrete search problem of three arcs. *SIAM Journal on Algebraic and Discrete Methods* 5(1):94–100.

[15] A. Dagan and S. Gal. Network search games, with arbitrary searcher starting point. *Networks* 52:156–161, 2008.

[16] S. Gal. Search games with mobile and immobile hider. *SIAM Journal on Control and Optimization* 17:99–122, 1979.

[17] S. Gal. *Search Games*. Academic Press, New York, 1980.

[18] S. Gal. On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory* 29:533–542, 2000.

[19] S. Gal. Search games. J. J. Cochran, ed. *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Hoboken, NJ, 2011.

[20] A. Garnaev. *Search Games and Other Applications of Game Theory*. Lecture Notes in Economics and Mathematical Systems, Vol. 485. Springer, New York, 2000.

[21] R. Isaacs. *Differential Games*. John Wiley & Sons, New York, 1965.

[22] A. Jotshi and R. Batta. Search for an immobile entity on a network. *European Journal of Operational Research* 191(2):347–359, 2008.

[23] K. Kikuta. A search game with travelling cost on a tree. *Journal of the Operations Research Society of Japan* 38(1):70–88, 1995.

[24] K. Kikuta and W. Ruckle. Initial point search on weighted trees. *Naval Research Logistics* 41(6):821–831, 1994.

[25] T. R. Lidbetter. Search games with multiple hidden objects. *SIAM Journal on Control and Optimization*. Forthcoming, 2013.

[26] L. Pavlovic. A search game on the union of graphs with immobile hider. *Naval Research Logistics* 42(8):1177–1199, 1995.

[27] J. H. Reijnierse and J. A. M. Potter. Search games with immobile hider. *International Journal of Game Theory* 21:385–394, 1993.