Thomas Lidbetter

Search games with multiple hidden objects

Article (Published version)
(Refereed)

Original citation:
Lidbetter, Thomas (2013) *Search games with multiple hidden objects.* SIAM Journal on Control and Optimization, 51 (4). pp. 3056-3074. ISSN 0363-0129

DOI: 10.1137/120893938

© 2013 Society for Industrial and Applied Mathematics

This version available at: http://eprints.lse.ac.uk/55103/

Available in LSE Research Online: November 2015

SIAM J. Control Optim.
Vol. 51, No. 4, pp. 3056–3074

SEARCH GAMES WITH MULTIPLE HIDDEN OBJECTS[*]

THOMAS LIDBETTER[†]

Abstract. We consider a class of zero-sum search games in which a Searcher seeks to minimize the expected time to find several objects hidden by a Hider. We begin by analyzing a game in which the Searcher wishes to find k balls hidden among $n > k$ boxes. There is a known cost of searching each box, and the Searcher seeks to minimize the total expected cost of finding all the objects in the worst case. We show that it is optimal for the Searcher to begin by searching a k-subset H of boxes with probability $\nu(H)$, which is proportional to the product of the search costs of the boxes in H. The Searcher should then search the $n - k$ remaining boxes in a random order. A worst-case Hider distribution is the distribution ν. We distinguish between the case of a *smart* Searcher who can change his search plan as he goes along and a *normal* Searcher who has to set out his plan from the beginning. We show that a smart Searcher has no advantage. We then show how the game can be formulated in terms of a more general network search game, and we give upper and lower bounds for the value of the game on an arbitrary network. For 2-arc connected networks (networks that cannot be disconnected by the removal of fewer than two arcs), we solve the game for a smart Searcher and give an upper bound on the value for a normal Searcher. This bound is tight if the network is a circle.

Key words. search game, network, discrete optimization

AMS subject classifications. 90B40, 91A25, 90B15, 91A05

DOI. 10.1137/120893938

1. Introduction.

The first problem we consider is that of a *Searcher* who wishes to find a number of objects (or balls) hidden among a set of discrete locations (or boxes), each of which has a designated *search cost*. The Searcher looks in the boxes one by one, paying the search costs associated with the boxes he looks in, until he has found all the balls. He wishes to minimize the total search cost of finding the balls in the worst case, so we view the problem as a zero-sum game between the Searcher and a malevolent *Hider* who wishes to maximize the total search cost. This is a natural problem to consider and one which we face on an everyday basis. For example, before leaving the house in the morning we may wish to locate certain essential items such as wallet, phone, and keys. There is a set of discrete locations around the house where these objects may be hidden, each of which takes a particular amount of time to search, and we wish to minimize the total time it takes to find all the items. The problem provides a simple model for other practical search scenarios, such as a search for a number of corrupted files which may be distributed among several folders, or a search for bombs hidden in several locations. The problem is also relevant to studies such as [23], which have examined how *scatter hoarders* like squirrels search for food they have previously cached over a number of sites.

In section 2.1, after we have formally defined the problem, we will see that if both parties are allowed to use randomized strategies it is optimal for the Searcher to begin his search with a subset of k boxes chosen with probability proportional to the product of their search costs and then search the remaining boxes in a random order. It is optimal for the Hider to choose a subset of k boxes with probability

[*] Received by the editors October 4, 2012; accepted for publication (in revised form) May 7, 2013; published electronically August 1, 2013.
http://www.siam.org/journals/sicon/51-4/89393.html

[†] Department of Mathematics, London School of Economics, WC2A 2AE London, UK (T.R.Lidbetter@lse.ac.uk).

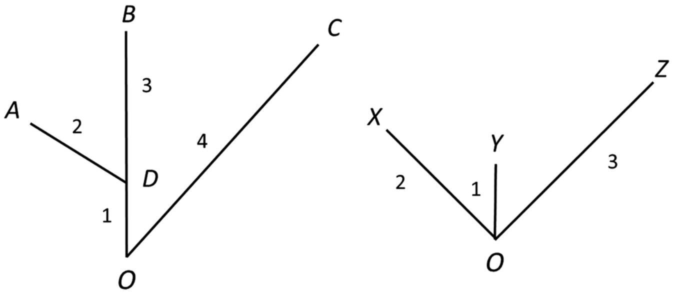

Fig. 1.1. *Two networks.*

proportional to the product of their search costs. In section 2.2 we then distinguish between two versions of the game, the first of which restricts the Searcher to setting out his search plan from the start, and the second of which allows him to change his plan during the search, based on information gathered. We call the second type of Searcher a *smart* Searcher to distinguish from the first type, which we call a *normal* Searcher. The idea of a smart Searcher was introduced in [6] in the context of scatter hoarders. A smart Searcher is clearly at an advantage over a normal Searcher, and this formulation perhaps provides a more realistic model of an intelligent Searcher. However, we will show that in this game a smart Searcher cannot do any better than a normal Searcher against a Hider who is playing optimally.

In section 3 we consider a continuous time version of the game described above. The Hider places k objects in some search space and the Searcher chooses a continuous time search strategy in this space to minimize the expected time taken to find all the objects. In particular we will take as our search space a network with given arc lengths, and the type of search we shall consider is known as *expanding search*, recently introduced in [8]. We define an expanding search of a network Q with root O as a sequence of unit speed paths, the first of which starts at O and the remainder of which each start from a point already reached by some previous path. The part of Q that has been searched by time t is then a subset of Q that increases, or expands with t.

To illustrate the principle of expanding search, consider the two networks depicted in Figure 1.1. The numbers denote the lengths of the arcs and the nodes are labeled with capital letters. An example of an expanding search on the network on the left is the sequence of paths OD, DA, DB, and OC. We will consider the *expanding search game* with k hidden objects on a network, in which the Searcher chooses an expanding search and the Hider chooses k points on the network. The payoff, which the Searcher wishes to minimize and the Hider to maximize, is the time taken until all the objects have been found. For the expanding search described above, if $k = 2$ and the Hider chooses the nodes A and B, the payoff is $1 + 2 + 3 = 6$.

As we explain in section 3.1, a search for objects in boxes, outlined at the beginning of this section, is equivalent to an expanding search for objects hidden on the terminal nodes of a star network. By a star network we mean a collection of arcs all meeting at a single point, O. The network on the right of Figure 1.1 is a star network, and an expanding search for objects hidden on the network among the nodes X, Y, and Z is equivalent to a search for objects hidden in three boxes with costs 2, 1, and 3. This means the box-searching game examined in section 2 can be generalized in terms of an expanding search for multiple objects hidden on a network. In section 3.1 we go on to discuss the game on the more general tree network and show that in general, a smart Searcher has an advantage over a normal Searcher. In section 3.2 we give some bounds on the expected search time in the expanding search game for multiple objects on general networks, and in section 3.3 we turn our attention to the game played on 2-arc connected networks. A 2-arc connected network is a network that cannot be disconnected by the removal of fewer than two arcs. We give the solution of the game for a smart Searcher and an upper bound for the value of the game for a normal Searcher, showing that although this bound is not tight in general, it is if the network is a circle.

The expanding search game on a network is a specific example of a more general class of *search games* on networks. Interest in search games was sparked by Isaacs' book on differential games [19], and accounts of the main problems and results in the field can be found in [11], [7], [16], and [15]. Expanding search is a variation on the classic form of a zero-sum search game on a network, in which a Hider picks a point on the network and a Searcher picks a time-minimizing unit speed path on the network, beginning from a designated root. The first examples of this game were solved in [13], with further results in [14]. More recently, variations on this game have been proposed, for instance in [3], where the network is supposed to have asymmetric travel times.

The box-searching game solved in section 2 is an original problem. The problem of searching for a *single* object hidden according to a known distribution in boxes with search costs where there is an overlook probability associated with each box is a known problem solved by Blackwell (reported in [20]). An alternative solution to the problem is presented in [17] using Gittins' well-known index for multiarmed bandit processes. A study of the zero-sum game version of the problem can be found in [12] and [27]. In [22], the game is extended so that the boxes each have allocated search costs. Interest in searching in boxes also extends to the field of economics, for example in [31], where a Searcher faces a problem of when to stop searching a set of boxes with fixed search costs, and rewards are assigned according to a known probability distribution.

To date there has been little study in search theory of problems involving multiple hidden objects. Alpern et al. [4] introduced a game in which a Searcher with limited digging resources seeks several objects buried in a choice of locations. An extension of Blackwell's problem in which a Searcher looks for *one* of many hidden objects is examined in [10] and [28]. In [21], two Searchers compete to find different objects before the other. Press [24] considers a search problem in which a Searcher samples with replacement to find "rare malfeasors" hidden according to a known distribution amongst a population. This could be viewed as a search for balls distributed among several boxes according to a known distribution. In [5], a search game is considered in which a Hider distributes a *continuous* amount of wealth among discrete locations.

The related problem of multiple Searchers trying to locate an object has been studied in [25], [26], and [29]. In these studies, two Searchers coordinate to find a single Hider. In [30] two agents aim to minimize the time to find each other in a set of discrete locations. This problem lies in the field of rendezvous search, a category of problems first posed informally in [1] and later developed in work such as [2] and [9].

2. Search for k balls in n boxes.

A Searcher wishes to find k balls hidden in a set $B = \{1, \ldots, n\}$ of n boxes. He can search them in any order but incurs a known cost c_i when he searches box i. If the balls are hidden (uniformly) randomly, then clearly the best search strategy is to search the boxes in order of increasing cost. We wish to find the randomized search strategy which minimizes the expected total cost, in the worst case. For any pure Searcher strategy—that is, any ordering (or permutation) of the boxes B—the worst case occurs when the last (nth) box searched contains a ball. So any pure search strategy is a pure minimax strategy with total cost equal to the sum of the search costs $\Sigma_{i=1}^{n} c_i$, which we denote by C_0.

We are naturally led to consider mixed search strategies and to seek a distribution over orderings of B which minimizes the expected total cost in the worst case. This problem is equivalent to finding optimal strategies in the zero-sum game $\Gamma = \Gamma(n, k; c_1, \ldots, c_n)$ against nature, where a malevolant Hider (nature) chooses a k-subset H of B, the Searcher chooses an ordering $(i_1, \ldots, i_n)$ of B, and the payoff (to the maximizing Hider) is the total cost

$$C = C(i_1, \ldots, i_n; H) = c_{i_1} + c_{i_2} + \cdots + c_{i_j}$$

if the last ball is found in the jth box to be searched. This is a finite game and has a value which we denote by $V = V(n, k; c_1, \ldots, c_n)$.

2.1. Optimal strategies. A Hider mixed strategy is a probability distribution over the set $\mathcal{H} = B^{(k)}$ of k-subsets (that is, subsets of size k) $H \subset B$. The optimal Hider distribution $\nu = \nu_{B,k}$ turns out to be the distribution which assigns to each $H \in \mathcal{H}$ a probability proportional to $\pi(H) := \Pi_{i \in H} c_i$, the product of the costs of searching the k boxes in H. That is,

$$\nu(H) = \frac{\pi(H)}{\sum_{A \in \mathcal{H}} \pi(A)}.$$

The distribution is related to a Hider distribution on trees determined recursively by Gal [13]. The main result of this section is the following.

THEOREM 2.1. *The Searcher strategy which minimizes the expected total search cost to find k balls among n boxes in the worst case is to first search a k-subset of boxes in any order, chosen according to the distribution ν, and then to search the remaining $n - k$ boxes in a (uniformly) random order. The order of searching the first k boxes does not affect the payoff. A worst case Hider distribution is the distribution ν.*

To prove this theorem, we will first show that the Hider strategy ν ensures the same expected search cost C against any strategy of the Searcher. This puts a lower bound on the value of the game. We then restrict the Searcher to a subset of his strategy set and show that even with this restriction he has a mixed strategy that ensures the same expected search cost C against any Hider strategy. This puts an upper bound on the value which when combined with the lower bound provides the desired result.

We begin by defining a quantity $S_j(H)$ for all subsets $H \subset B$ and for all $j = 1, \ldots, |H|$. $S_j(H)$ is calculated by taking each j-subset A of H, multiplying together the search costs of the boxes in A, and summing these products.

DEFINITION 2.2. *For $H \subset B$ and $j = 1, \ldots, |H|$, let $S_j(H) = \sum_{A \in H^{(j)}} \pi(A)$. We write simply S_j for $S_j(B)$.*

Note that this allows us to notate the Hider strategy ν more concisely: for k-subsets H, we can write $\nu(H) = \pi(H)/S_k$. We will show that ν makes the Searcher indifferent between all his pure strategies, ensuring the same expected cost for any ordering of the boxes. Following standard notation, we write $[j]$ for the set $\{1, \ldots, j\}$ so that $B = [n]$.

LEMMA 2.3. *For any ordering $(i_1, \dots, i_n)$ of the boxes, the expected total search cost if the balls are hidden according to ν is*

$$C = C(i_1, \dots, i_n; \nu) = C_0 - \frac{S_{k+1}}{S_k}. \tag{2.1}$$

Proof. By a relabeling argument we can assume that the Searcher chooses the ordering $1, 2, \dots, n$. We calculate the expected cost of boxes not searched and subtract this from the total cost of the boxes, C_0. All the boxes $1, \dots, k$ will certainly be searched, and for $i \geq k+1$, the probability box i is not searched is the probability all k balls are in $[i-1]$, which is

$$\frac{S_k\left([i-1]\right)}{S_k}.$$

Thus the expected cost of boxes not searched is

$$\frac{1}{S_k} \sum_{i=k+1}^{n} c_i S_k\left([i-1]\right).$$

This sum is clearly equal to S_{k+1}, so the expression above is S_{k+1}/S_k, and subtracting this from C_0 gives (2.1). □

We will see that the Hider strategy ν set out at the beginning of the section is optimal and that the value of the game is given by (2.1). To this end, we now define a restricted game $\Gamma' = \Gamma^{'}(n, k; c_1, \dots, c_n)$ in which the Searcher's strategy set is reduced, and we will show that in Γ' the Searcher can attain expected search time $C_0 - S_{k+1}/S_k$ against any Hider strategy.

DEFINITION 2.4. *Let Γ' be the same as Γ except that after searching the first k boxes the Searcher must search the remaining boxes in a (uniformly) random order. Let $V' = V'(n, k; c_1, \dots, c_n)$ be the value of this game. We must have $V' \geq V$.*

This reduces the number of pure strategies for the normal Searcher to $\binom{n}{k}$, since a pure strategy can now be specified by some $A \in B^{(k)}$ which corresponds to the first k boxes of his search. Hence the Hider and the Searcher have the same strategy set, $\mathcal{H} = B^{(k)}$. We now show that Γ' is a symmetric game in the sense that its payoff matrix is symmetric.

LEMMA 2.5. *Let $H, A \in \mathcal{H}$ be strategies in Γ' for the Hider and Searcher, respectively. Then $C(A, H) = C(H, A)$.*

Proof. $C(A, H)$ is the expected cost of a search that begins with A and searches the rest of the boxes in a random order until all the balls in H are found. In such a search all the boxes in $A \cup H$ must be searched before the balls have all been found, since the Searcher begins by searching all the boxes in A and cannot stop until he has searched all the boxes in H. As for the remaining boxes not in $A \cup H$, these are each searched with the same probability q after all the boxes in A have been searched. Note that the value of q depends on only two things. The first is the number $j = |A - H|$ of remaining balls to be found after the boxes in A have been searched, and the second is the number $m = n - k$ of remaining unsearched boxes at this point. Hence

$$C(A, H) = \sum_{i \in A \cup H} c_i + q \sum_{i \notin A \cup H} c_i.$$

Notice that $|A - H| = |H - A|$, and so j, m, and q are unchanged if the set A and H are interchanged. Thus $C(A, H) = C(H, A)$. □

The solution of the game Γ' follows immediately from this lemma, using the well-known result below about finite zero-sum games with symmetric payoff matrices.

LEMMA 2.6. *For a two-player, zero-sum game between Players* I *and* II *with* $n \times n$ *symmetric payoff matrix* M, *if* $x^* = (x_1^*, \dots, x_n^*)^T$ *is a mixed strategy for Player* I *that makes Player* II *indifferent between all his strategies, then the strategy pair* (x^*, x^*) *forms an equilibrium.*

Proof. Since x^* makes Player II indifferent, there is some number U such that $U = (x^*)^T My$ for all strategies y of Player II. Equivalently,

$$\begin{aligned} U^T = U &= \left((x^*)^T My\right)^T \\ &= y^T M^T \left((x^*)^T\right)^T \\ &= y^T M x^*, \text{ since } M \text{ is symmetric.} \end{aligned}$$

Since this holds for all y, Player II can make Player I indifferent between all his strategies by playing x^*. Hence if both players play x^* each player is playing a best response to the other, and this is an equilibrium. □

COROLLARY 2.7. *The value of* Γ' *is* $V^{'} = C_0 - \frac{S_{k+1}}{S_k}$. *The strategy* ν *is optimal for both the Hider and the Searcher.*

Proof. If the Hider uses the strategy ν, by Lemma 2.3, the Searcher will be indifferent and the expected cost will be $C_0 - \frac{S_k}{S_{k+1}}$. By Lemma 2.5 Γ' is symmetric, so by Lemma 2.6 if the Searcher also uses the mixed strategy ν, this forms an equilibrium and the value of the game is $V^{'} = C(\nu, \nu) = C_0 - \frac{S_{k+1}}{S_k}$. □

Theorem 2.1 follows by combining our results.

Proof of Theorem 2.1. By Lemma 2.3, the Hider can ensure an expected search cost of no less than $C_0 - \frac{S_{k+1}}{S_k}$ by using ν, so $V \geq C_0 - \frac{S_{k+1}}{S_k}$. By Corollary 2.7, the Searcher can ensure an expected search cost of no more than $C_0 - \frac{S_{k+1}}{S_k}$ by using the strategy ν, so $V \leq V^{'} = C_0 - \frac{S_{k+1}}{S_k}$. Hence we must have equality, and ν is optimal for both players. □

Note that the value of the game must be increasing in k since the Searcher's optimal strategy for the game $\Gamma(n, k+1; c_1, \dots, c_n)$ will find k hidden balls with total search cost no greater than $V(n, k+1; c_1, \dots, c_n)$. Therefore we must have

$$\begin{aligned} C_0 - \frac{S_{k+1}}{S_k} &\geq C_0 - \frac{S_k}{S_{k-1}} \text{ so that} \\ S_k^2 &\geq S_{k-1}S_{k+1} \text{ and} \\ \log S_k &\geq \frac{\log S_{k-1} + \log S_{k+1}}{2}. \end{aligned}$$

It follows that $S_1, S_2, \dots$ is a logarithmically concave sequence. We can also see this in another way using the concept of logarithmically concave polynomials (that is, polynomials whose sequence of coefficients is logarithmically concave). It is well known that the product of logarithmically concave polynomials is logarithmically concave [18]. That is, if $A(x) = a_0 + a_1 x + a_2 x^2 + \cdots$ and $B(x) = b_0 + b_1 x + b_2 x^2 + \cdots$ are logarithmically concave so that they satisfy $a_k \geq a_{k+1}a_{k-1}$ and $b_k \geq b_{k+1}b_{k-1}$ for all $k \geq 1$, then $C(x) = A(x)B(x)$ is logarithmically concave. Let $S(x)$ be the polynomial whose coefficients are the S_k, so

$$S(x) = 1 + S_1 x + S_2 x^2 + \cdots + S_n x^n.$$

Then we can factorize $S(x)$ as

$$S(x) = \prod_{j=1}^{n} (1 + c_j x),$$

and since the polynomial $\left(1 + c_j x + 0x^2 + 0x^3 + \cdots\right)$ is logarithmically concave for each j, $S(x)$ is the product of logarithmically concave polynomicals and must be logarithmically concave itself.

2.2. Smart and normal strategies. So far we have assumed that the Searcher must set out his search plan at the start of the search, but we now consider a variation of the game where the Searcher is permitted to adapt his plan during the search using information gathered. We call such a Searcher smart, as opposed to a normal Searcher, who is required to set out his search plan from the start. In the case of a single hidden ball it is clear that a smart Searcher cannot do any better than a normal Searcher. In Alpern et al. [6] the authors showed that for their model of scatter hoarding behavior it was advantageous for a Searcher to be smart. However, we find the contrary here, that a smart Searcher has no advantage over a normal Searcher.

We denote the box-searching game with a smart Searcher by $\tilde{\Gamma} = \tilde{\Gamma}(n, k; c_1, \ldots, c_n)$ and let the value of the game be $\tilde{V} = \tilde{V}(n, k; c_1, \ldots, c_n)$. Clearly the value of this game is no greater than the value of the equivalent game with a normal Searcher, since a smart Searcher can always use a normal strategy. We combine this observation with Theorem 2.1 to give the lemma below.

LEMMA 2.8. *$\tilde{V} = \tilde{V}(n, k; c_1, \ldots, c_n)$ satisfies $\tilde{V} \leq C_0 - \frac{S_{k+1}}{S_k}$.*

To show that a smart Searcher cannot do better, we will need a technical lemma about the function S_k, which is a straightforward calculation. For any set $H \subset B$ write $S_j^i = S_j(B - i)$.

LEMMA 2.9. *For any $i = 1, \ldots, n$,*

$$S_k = c_i S_{k-1}^i + S_k^i.$$

Proof. We split the sum $S_k = \sum_{A \in B^{(k)}} \pi(A)$ into the sum over subsets A which include i and those that do not. Accordingly,

$$\begin{aligned} S_k &= \sum_{A \in (B-i)^{(k)}} \pi(A) + \sum_{A \in (B-i)^{(k-1)}} c_j \pi(A) \\ &= c_i S_{k-1}^i + S_k^i. \qquad \square \end{aligned}$$

We use this lemma to give an inductive proof that a smart Searcher strategy cannot do any better than a normal Searcher strategy against ν. The idea of the proof is that the information gathered during a search is of no use to the Searcher because ν has the property that at all points of a search, the remaining balls to be found will be hidden according to the optimal Hider distribution on the unsearched boxes. We formalize this in the next lemma.

LEMMA 2.10. *Suppose the k balls are hidden according to the optimal Hider distribution ν for the game $\Gamma(n, k; c_1, \ldots, c_n)$, and let $i \in B$. Let $\nu' = \nu_{B-i,k-1}$ be the optimal Hider distribution for $k-1$ balls hidden in the boxes $B - i$ and let $\nu'' = \nu_{B-i,k}$ be the optimal Hider distribution for k balls in the boxes $B - i$. Then given that there* *is a ball in box i, the remaining $k-1$ balls are hidden in $B-i$ according to ν'; given that there is no ball in box i, the k balls are hidden in $B-i$ according to ν''.*

Proof. The probability α that there is a ball in box i is given by $\alpha = \frac{c_i S^i_{k-1}}{S_k}$. Hence, conditional on there being a ball in box i, the probability the remaining $k-1$ balls are hidden in some $(k-1)$-set $H \in (B-i)^{(k)}$ is

$$\frac{\nu(H \cup i)}{\frac{c_i S^i_{k-1}}{S_k}} = \frac{\frac{\pi(H)c_i}{S_k}}{\frac{c_i S^i_{k-1}}{S_k}} = \frac{\pi(H)}{S^i_{k-1}} = \nu'(H).$$

Similarly, the probability there is no ball in box i is $1-\alpha = 1 - \frac{c_i S^i_{k-1}}{S_k} = \frac{S_k - c_i S^i_{k-1}}{S_k} = \frac{S^i_k}{S_k}$ (by Lemma 2.9). Hence, conditional on there not being a ball in box i, the probability the remaining k balls are hidden in some k-set $H \subset B-i$ is

$$\frac{\nu(H)}{\frac{S^i_k}{S_k}} = \frac{\frac{\pi(H)}{S_k}}{\frac{S^i_k}{S_k}} = \frac{\pi(H)}{S^i_k} = \nu''(H). \qquad \square$$

From this property of ν it follows that a smart Searcher cannot guarantee a total search cost any smaller than a normal Searcher.

THEOREM 2.11. *The value of $\tilde{\Gamma}$ is $\tilde{V} = C = C_0 - \frac{S_{k+1}}{S_k}$. The strategy ν is optimal for both the Hider and the Searcher.*

Proof. We prove by induction on n that if the balls are hidden according to ν, then the expected search cost of any smart search is C. This is clearly true for $n = 2$, since every smart search is a normal search. Assume it is true for $n-1$, and suppose without loss of generality that a smart Searcher begins by searching box 1. Subsequent to searching this box, regardless of whether he finds an object, Lemma 2.10 implies that the remaining objects will be hidden optimally. Hence, by the induction hypothesis and Lemma 2.3 the expected search cost of all smart searches of the remaining boxes is the same, including the search which opens the boxes in the order $2, 3, \dots, n$. So the total expected search cost is the same as that of the normal search $1, 2, \dots, n$, which by Lemma 2.3 is C, completing the induction. It follows that $\tilde{V} \geq C$, and combining this with Lemma 2.8 completes the proof. $\square$

We can also consider variations of the game in which the Hider is smart, so that every time the Searcher opens a box, he can rearrange the remaining balls that have not been found. This gives rise to two more games, one in which the Searcher is also smart and one in which he is normal. The value of these games is clearly greater than the values of the corresponding games in which the Hider is normal, which we have seen are both equal to $C = C_0 - \frac{S_{k+1}}{S_k}$.

In the case where only the Hider is smart, the value is strictly greater than C in general, though the game is nontrivial to analyze and there are no examples for which the value is greater than C when $n \leq 3$. We present here an example for $n = 4$. Suppose the boxes $(1, 2, 3, 4)$ have search costs $(1, 10, 50, 50)$ and there are two balls. We give a Hider strategy that ensures expected search cost greater than C for any normal Searcher strategy. The Hider starts by hiding the balls according to the optimal normal strategy ν. If the Searcher opens box 1 first and finds a ball, the Hider puts the remaining ball in box 2. If not, he hides the two remaining balls in boxes 3 and 4 with probability $197/245$ and otherwise hides them equiprobably in boxes 2 and 3 or boxes 2 and 4. If the Searcher starts with box 2 and finds a ball, the Hider puts the remaining ball in box 1 with probability $361/10201$ and otherwise hides them equiprobably in box 3 or 4. If not, he puts the other two balls in boxes 3 and 4. If the Searcher starts with box 3 or 4 and finds a ball, the Hider puts the remaining ball in boxes 1 or 2 with respective probabilities 361/18605 and 9122/55815 and otherwise puts it in the other box of cost 50. If not, he puts the two balls in box 2 and the remaining box of cost 50. We leave it as an exercise to the reader to check that against any normal Searcher strategy the expected cost is greater than C.

If both the Hider and the Searcher are smart, then it turns out the value is C. To prove this, it is sufficient to give a strategy for a smart Searcher that makes the Hider indifferent between all his strategies, since we already know by Theorem 2.11 that the Hider strategy ν ensures an expected search cost of C against any smart search. To define a smart search we simply need to specify the probability p_i that the Searcher begins with some box $i = 1, \dots, n$ for given n and k. Let

$$p_i = \left(\frac{S_k^i}{S_{k-1}^i} - \frac{S_{k+1}^i}{S_k^i} \right) \left(\sum_{j=1}^n \frac{S_k^j}{S_{k-1}^j} - \frac{S_{k+1}^j}{S_k^j} \right)^{-1}.$$

Since S_k^i is a logarithmically concave sequence, $\left(S_k^i/S_{k-1}^i - S_{k+1}^i/S_k^i\right)$ is non-negative, so p_i is certainly a probability. We then prove by induction on n that when the Searcher uses p, the expected search cost of boxes not searched is the same whatever strategy the Hider uses, so that it must be equal to S_{k+1}/S_k. The base case $n = 2$ is easily verified. Assuming the induction hypothesis is true for $n-1$, the expected search cost against the Hider strategy H is

$$\sum_{i \in H} p_i \frac{S_k^i}{S_{k-1}^i} - \sum_{i \notin H} p_i \frac{S_{k+1}^i}{S_k^i}. \tag{2.2}$$

It is sufficient to show that expected search cost is the same against any two Hider strategies H and H' that differ only in that H contains i and H' contains j. The difference in the quantity (2.2) for two such Hider strategies is

$$p_i \left(\frac{S_k^i}{S_{k-1}^i} - \frac{S_{k+1}^i}{S_k^i} \right) - p_j \left(\frac{S_k^j}{S_{k-1}^j} - \frac{S_{k+1}^j}{S_k^j} \right),$$

which is 0, by our choice of p. Summing this up, we have the following.

THEOREM 2.12. *The value of the game with a smart Searcher and smart Hider is C. The strategy ν is optimal for the Hider and the strategy p given above is optimal for the Searcher.*

3. Expanding search for *k* objects on a network.

We can reformulate the games analyzed in section 2 in terms of an expanding search game. Recall from the introduction that an expanding search of a network Q with root O is a sequence of unit speed paths, the first of which starts at O and the remainder of which each start from a point already reached by some previous path.

In this section we begin by describing how searching in boxes can be thought of as a special case of expanding a search on a network, and we define precisely the expanding search game for multiple objects on a network. We then go on to examine this game, focusing particularly on 2-arc connected networks in section 3.3.

3.1. Box search as expanding search on a network.

Consider the star network Q_n consisting of n arcs $a_1, \dots, a_n$ which meet at the root O, as depicted in Figure 3.1.

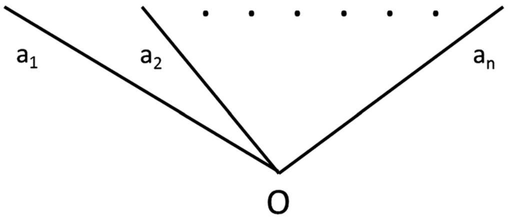

FIG. 3.1. *The network* Q_n.

For given costs $c_1, \ldots, c_n$, let arc a_i have length c_i. We can now define a search game $\Lambda(Q_n, k)$ in which a normal Searcher chooses an expanding search starting at O and the Hider chooses k points on Q_n. An expanding search on Q_n is essentially an ordering of the arcs $a_1, \ldots, a_n$. We define the payoff of the game as the first time all the points chosen by the Hider have been reached by the search. It is optimal for the Hider to choose k distinct leaf nodes (that is, the tips of the arcs), and so it is easy to see that $\Lambda(Q_n, k)$ is equivalent to the box-searching game $\Gamma(n, k; c_1, \ldots, c_n)$, which we have solved in section 2. Of course, for any network Q with root O we can define in the natural way an associated expanding search game, $\Lambda(Q, k)$ with k hidden objects, where a Hider chooses k points on Q and a normal Searcher chooses an expanding search of Q starting at O. We denote the value of the game (if it exists) by $V = V(Q, k)$.

For $k = 1$ the solution of the game $\Lambda(Q, k)$ can be found in [8] for general tree networks Q, and a formula is given for the value. This formula is $V = (\mu + D)/2$, where μ is the total length of the network and D is an average of the distances of the root to the leaf notes, weighted with respect to the optimal Hider distribution on the leaf nodes. For a star, $\mu = C_0$ and $D = \sum \frac{c_i}{C_0} c_i = \frac{1}{C_0} \sum c_i^2$. Our formula from Theorem 2.1 gives

$$\begin{aligned} V &= C_0 - \frac{S_2}{S_1} \\ &= C_0 - \frac{\sum_{i<j} c_i c_j}{C_0} \\ &= \frac{C_0}{2} + \frac{C_0^2 - 2\sum_{i<j} c_i c_j}{2C_0} \\ &= \frac{C_0}{2} + \frac{\sum c_i^2}{2C_0} \\ &= \frac{1}{2}(C_0 + D). \end{aligned}$$

This shows that the two formulas are equivalent for $k = 1$. The optimal Hider strategy given in [8] is also the same, but the optimal Searcher strategy is different. It mixes between 2^{n-1} pure strategies, whereas our optimal strategy mixes between all $n!$ pure strategies.

For the game $\Lambda(Q, k)$ played on general trees Q, it is clear that it is optimal for the Hider to place the objects at distinct (if possible) leaf nodes of the tree. A Searcher pure strategy is a sequence of arcs of the tree, the base of each of which touches one of the arcs already searched. Λ can therefore be reduced to a finite game and must have a value. In a small number of special cases, the solution of the game on a tree can be deduced from the solution of the game on a star. For example, consider the tree

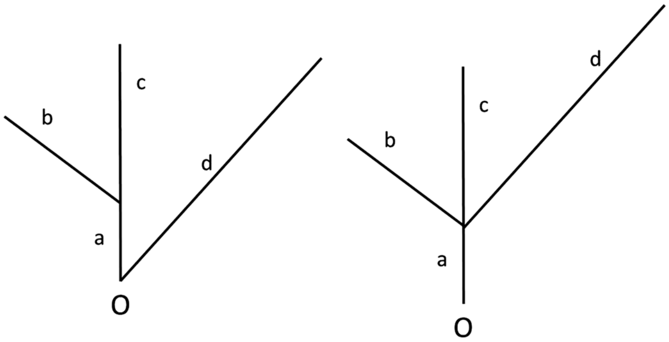

Fig. 3.2. *Two tree networks.*

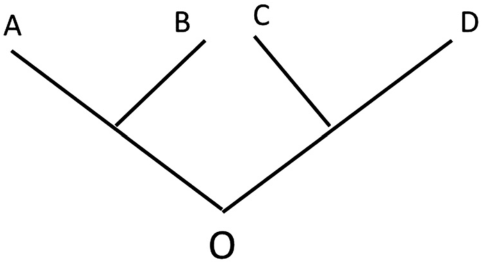

Fig. 3.3. *The network Q.*

depicted on the left of Figure 3.2. For the game with two hidden objects, it is clear that the arc a must be traversed by the Searcher before he finds all the hidden objects, since there must be an object at the leaf node of arc b or c. Hence this network has the same value as the network depicted on the right of Figure 3.2, since the Searcher may as well begin by traversing the arc a. The solution of the game on the network on the right follows easily from the solution of the game on a star network.

This principle can be extended to more general trees but does not produce strong results. If a tree is binary (each branch node has two outward arcs), then the solution of the search game for k Hiders can be deduced from the solution on a star only if there are no more than $k + 1$ leaf nodes.

We investigate the solution of the game for $k = 2$ on a particular tree Q, depicted in Figure 3.3, with leaf nodes A, B, C, D. All arcs have unit length. Up to symmetry, the Hider has only two pure strategies: hide on the same side (choose nodes AB or CD) or hide on different sides (choose nodes AC, AD, BC, or BD). Similarly, the Searcher has only two pure strategies up to symmetry: start with two nodes on the same side or start with two nodes on different sides. Hence the game can be reduced to a 2×2 matrix game in which the Hider chooses between the strategies *same* and *diff* and the Searcher chooses between *SAME* and *DIFF*, the interpretation being that if, say, the Hider chooses *same*, he randomizes between AB and AC. This gives the following representation of the game in strategic form:

Hider/Searcher	*SAME*	*DIFF*
same	4.5	5.5
diff	5.5	5.25

Solving this gives a value of $V = 5.3$ with both players' unique optimal strategies given by the probability vector $(1/5, 4/5)$. Note that in general the game played on a tree does not have a symmetric payoff matrix.

As before, we can also define the related game $\tilde{\Lambda}(Q, k)$ on a network Q with root O, in which the Hider chooses k points on the network and a *smart* Searcher chooses an expanding search of Q starting at O, so that at any point during the search the Searcher is permitted to change his search plan. We denote the value of this game (if it exists) by $\tilde{V} = \tilde{V}(Q, k)$, which can be no greater than V. If Q is a tree, both a smart and a normal Searcher have a finite strategy set, so $V(Q, k)$ and $\tilde{V}(Q, k)$ both certainly exist.

Theorems 2.1 and 2.11 show that if Q is a star network, then $V(Q, k) = \tilde{V}(Q, k)$. We now show that this may not be true for general trees Q. Consider the smart search game $\tilde{\Lambda}(Q, 2)$ played on the tree Q in Figure 3.3 with $k = 2$, and suppose a smart Searcher uses the following strategy. He searches the nodes in the order $ACDB$ if there is an object at A, and otherwise he searches them in the order $ABCD$. If the Hider uses the strategy $(1/5, 4/5)$ which is uniquely optimal in the game $\Lambda(Q, k)$, then the Searcher will find an object at A with probability $1/2$. If he does find an object here, then with probability $4/5$ he finds the remaining object at one of the nodes on the other side after total expected time 4.5; with probability $1/5$ he finds the remaining object only after searching the whole network at time 6. So if he finds an object at A the expected search time is $4/5(4.5) + 1/5(6) = 4.8$. Similarly, if he doesn't find an object at A, then with probability $4/5$ he finds an object at B and the remaining object after expected time 5.5; with probability $1/5$ he finds both objects on the other side after time 6. So if he doesn't find an object at A the expected search time is $4/5(5.5) + 1/5(6) = 5.6$. So on average the expected search time is $1/2(4.8) + 1/2(5.6) = 5.2 < V(Q, 2)$.

We have showed that against the Hider strategy $(1/5, 4/5)$ there is a smart search that guarantees expected search time strictly less than $V(Q, 2)$. If the Hider follows any other strategy, then this must be suboptimal in the normal search game $\Lambda(Q, 2)$, so there is a normal search (and hence a smart search) which guarantees expected time strictly less than $V(Q, 2)$. Hence by the minimax theorem for zero-sum games, the value $\tilde{V}(Q, 2)$ of the smart search game $\tilde{\Lambda}(Q, 2)$ satisfies $\tilde{V}(Q, 2) < V(Q, 2)$.

It can be shown that up to symmetry a smart Searcher has six pure strategies in the game $\tilde{\Lambda}(Q, 2)$, and the value is $\tilde{V} = 5.25$. The optimal Hider strategy is $(1/4, 3/4)$ and the optimal Searcher strategy can be described as follows. Pick a node at random to start with. If there is no object here, then search the other node on the same side and the remaining two in a random order. If there is an object here, with probability $1/2$ search the other node on the same side and the remaining two in a random order, and with probability $1/2$ search the two nodes on the other side first then the remaining node.

3.2. Some strategies for arbitrary networks. For the games $\Lambda(Q, k)$ and $\tilde{\Lambda}(Q, k)$ played on a general network the Hider may no longer have a finite set of dominating strategies as he does on a tree, so that Λ and $\tilde{\Lambda}$ are infinite games. In this case it is not immediately obvious they have a value. In this section we give upper and lower bounds for the values of the games, if they exist, on an arbitrary network.

We first introduce some notation to describe an expanding search of a network Q with root O and total length μ. For a given expanding search (that is a sequence of unit speed paths starting from O, each of which starts from a point already reached), let $P(t)$ $(t \geq 0)$ denote the unique point in Q reached by the search at time t. We must have $P(0) = O$ and $P([0,t])$ must be a connected set of measure at most t for all $t \geq 0$. For an expanding search P and a set H of k points in Q, we denote the time taken for P to find all k points in H by $T(P,H)$.

The set of expanding searches is a subset of a larger class of search strategies, which in [7] are called *generalized search strategies.* These can be defined in an arbitrary search space Q, which may not be a network but could be any subset of Euclidean space. The Lebesque measure of a measurable subset $X \subset Q$ is denoted by $\mu(X)$ with the total measure of Q denoted by $\mu = \mu(Q)$.

DEFINITION 3.1. *A generalized search strategy in a search space Q is defined by the sets $S(t) \subset Q$ that have been "discovered" by time t. The sets $S(t)$ are required to satisfy the following conditions:*

$$S(t) \subset S(t') \textit{ for } t < t' \textit{ and } \mu(S(t)) \leq t \textit{ for all } t \geq 0.$$

In particular, an expanding search $P(t)$ of a network is a generalized search strategy S, where $S(t) = P([0,t])$, satisfying the additional condition that $S(t)$ is connected for all t. We can also define a generalized *smart* search strategy as a generalized search strategy which the Searcher is permitted to adapt at any point during the search.

For an arbitrary search space Q we define a Hider strategy as a set of k points $H_1, \ldots, H_k$ in Q. We define below a mixed Hider strategy u_k which we call the *k-uniform strategy.* This is the strategy that places each of the k objects on Q independently and completely randomly.

DEFINITION 3.2. *The k-uniform strategy u_k on a search space Q is a set of randomly chosen points $h_1, \ldots, h_k$ in Q, each of which is independently chosen uniformly, so that for any measurable set $X \subset Q$, the probability h_i is in X is $\mu(X)/\mu$.*

Hiding according to the k-uniform strategy allows the Hider to put the following upper bound on the expected search time of any generalized smart search strategy. This generalizes an analagous result for a single Hider first noted in [19], later used in [13], and stated in more generality in [7].

THEOREM 3.3. *If the Hider chooses his hiding points according to the k-uniform strategy u_k, then he ensures an expected search time of at least $(1-1/(k+1))\mu$ against any generalized smart search strategy.*

Proof. Since the k objects are independently hidden, the times at which they are found by any generalized smart search strategy are independent. Hence a smart Searcher has no advantage over a normal Searcher, so we need only consider generalized search strategies that are not smart. For any such strategy $S(t)$, the probability that the time T it takes to find all k objects is less than some $t \geq 0$ is $(\mu(S(t))/\mu)^k$, by definition of u_k. But by definition of a generalized search strategy, $\mu(S(t)) \leq t$ for all $t \geq 0$, so

$$P(T < t) \leq \min\left[\left(\frac{t}{\mu}\right)^k, 1\right].$$

Hence the expected search time $E(T)$ is given by

$$\begin{aligned} E(T) &= \int_0^\infty P(T \geq t)dt \leq \int_0^\infty \max\left[1 - \left(\frac{t}{\mu}\right)^k, 0\right] dt \\ &= \int_0^\mu \left(1 - \left(\frac{t}{\mu}\right)^k\right) dt = \left(1 - \frac{1}{k+1}\right)\mu. \qquad \square \end{aligned}$$

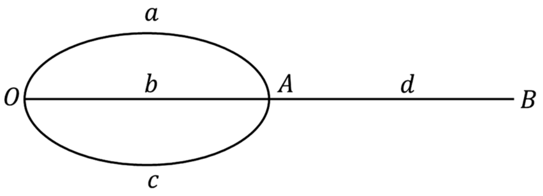

Fig. 3.4.

Theorem 3.3 gives a lower bound on the value (if it exists) of the expanding search game for k objects on any network with a smart Searcher. We will now give an upper bound by defining a smart search strategy for the game $\tilde{\Lambda}(Q,k)$ on any network Q.

We first introduce the notion of a *reversible* expanding search. Suppose $P(t)$ is an expanding search of a network Q for which τ is the first time that all the points of Q have been searched. We can define a function $P^{-1}(t)$, $0 \leq t \leq \tau$ such that $P^{-1}(t) = P(\tau - t)$. By a reversible expanding search we mean an expanding search P for which P^{-1} is also an expanding search. Note that not all expanding searches are reversible, for instance, those that don't end at the root. A necessary condition for P to be reversible is that for all t, $Q - P([0,t])$ is connected and contains O. Every network has a reversible expanding search of minimum length (that is, for which τ is minimal).

Definition 3.4. *A reversible expanding search that visits all the points of Q and has minimal length will be called a minimal reversible expanding search (MRES). We denote its length by $\bar{\mu} = \bar{\mu}(Q)$, which must be at least μ.*

It may be the case that a network's MRES doubles back on itself, as in the network in Figure 3.4, in which all the arcs, a, b, c, and d have unit length. An example of an MRES on this network can be described as follows: take the path along arc a from O to A, then the path along b from O to A, then the path from A to B and back again, then the path from A to O along c. Here the arc d is necessarily traversed twice, so that the length $\bar{\mu}$ of the MRES is 5.

We use the concept of the MRES to define a mixed strategy for a smart Searcher, called the smart k-uniform MRES.

Definition 3.5. *Let P be an MRES on a network Q, and for each $j = 0, 1, \ldots, k$ let P_j be the smart search that follows P until j objects have been found, then follows P^{-1} until the remaining $k - j$ objects have been found. The smart k-uniform MRES P_* is defined as an equiprobable choice of the P_j.*

Following P_* gives an upper bound on the expected search time, as we now show.

Theorem 3.6. *If a smart Searcher follows a smart k-uniform MRES P_*, he ensures an expected search time of no more than $\left(1 - \frac{1}{k+1}\right)\bar{\mu}$ against any Hider strategy H in the game $\tilde{\Lambda}(Q,k)$.*

Proof. Suppose the MRES P finds the k objects at times $t_1, t_2, \ldots, t_k$, where $t_1 \leq t_2 \leq \cdots \leq t_k$, and let $t_0 = 0$ and $t_{k+1} = \bar{\mu}$. Then for $j = 0, 1, \ldots, k$, the expanding search P_j finds j objects after time t_j, and after further time $\bar{\mu} - t_{j+1}$ will have found all the remaining objects. (In fact the remaining objects may be found by an earlier point in time.) So the total time taken for P_j to find all the objects is no greater than $t_j + (\bar{\mu} - t_{j+1})$. Since P_* is an equiprobable choice of the P_j, the total expected search time $T(P_*, H)$ satisfies

$$\begin{aligned} T(P_*, H) &\leq \frac{1}{k+1} \sum_{j=0}^{k} t_j + (\bar{\mu} - t_{j+1}) \\ &= \frac{1}{k+1} \left((k+1)\bar{\mu} - \sum_{j=0}^{k} (t_{j+1} - t_j) \right) \\ &= \frac{1}{k+1} \left((k+1)\bar{\mu} - \bar{\mu} \right) \\ &= \left(1 - \frac{1}{k+1} \right) \bar{\mu}. \qquad \square \end{aligned}$$

For the network in Figure 3.4, for $k = 3$, we have $(1-1/(k+1))\bar{\mu} = 5(3/4) = 3.75$ and $(1 - 1/(k + 1))\mu = 4(3/4) = 3$, so Theorems 3.3 and 3.6 imply that the value of the smart search game for three objects on this network (if it exists) is between 3 and 3.75.

We can also define a mixed strategy for a normal Searcher on any network, giving us an upper bound for the expected search time in the normal game.

Definition 3.7. *Let P be an MRES on a network Q, and let I_j be the portion of this MRES that is traversed during the time interval $\left[\frac{j-1}{k}\bar{\mu}, \frac{j}{k}\bar{\mu}\right)$, $j = 1, \ldots, k$. There is an expanding search for which I_j is the last part of Q to be searched, and it is searched in the backward direction, namely, the search which starts by following P until time $\frac{j-1}{k}\bar{\mu}$ and then follows P^{-1} for the remaining time. Similarly there is a search for which I_j is the last part of Q to be searched, and in the forward direction, which starts by following P^{-1} until time $\left(1 - \frac{j}{k}\right)\bar{\mu}$ and then follows P for the remaining time. The normal k-uniform MRES P^* is defined as the search strategy which makes an equiprobable choice between all $2k$ expanding searches so described.*

Theorem 3.8. *If a normal Searcher follows a normal k-uniform MRES P^*, he ensures an expected search time of no more than $\left(1 - \frac{1}{2k}\right)\bar{\mu}$.*

Proof. First note that if the MRES P doubles back on itself for any part of Q, then we can simply add some more nodes and arcs to Q to create a new network which has the same $\bar{\mu}$ but in which the MRES never doubles back. To do this, we simply add a new arc whenever P starts to retrace some part of Q that has already been searched. It is clear that this is a disadvantage to the Searcher, so it will only increase the expected search time. Without loss of generality we can restrict the proof to Q for which the MRES traverses each arc of Q only once, so that $\bar{\mu} = \mu$.

By rescaling we can assume $\bar{\mu} = 1$ so each I_j has length $1/k$. Supposing that I_j is the last part of Q to be searched by P^*, it is equally likely to be searched in the forward direction (by P) as it is in the reverse direction (by P^{-1}). It follows that if I_j contains just one object, then the expected search time will be $1 - (1/2)(1/k)$. If I_j contains no objects, then by the time I_j is searched all the objects will have been found and the search time is no more than $1 - 1/k$. If I_j contains more than one object, then the search time may be as great as 1. However, as the number of I_j containing two or more objects must be no greater than the number of I_j containing no objects, the expected search time is bounded by the average of $1 - 1/k$ and 1, which is $1 - (1/2)(1/k)$. $\square$

For the network in Figure 3.4, for $k = 3$, we have $\bar{\mu}(1-1/(2k)) = 5(5/6) \approx 4.167$, so Theorems 3.3 and 3.6 imply that the value of the normal search game for three objects on this network (if it exists) is between 3 and 4.167.

When $k = 1$, the two lower bounds given in Theorems 3.6 and 3.8 are both equal to $\bar{\mu}/2$. This bound for the expanding search game with one hidden object can be found in [8], and our results therefore both generalize this bound.

3.3. Search for k objects on a 2-arc connected network. In this section, we examine when the two bounds given in Theorems 3.3 and 3.6 are equal, that is, when $\mu(Q) = \bar{\mu}(Q)$. For this equality to hold, we must be able to find an MRES on Q which doesn't double back on itself. We will see that this is possible precisely for 2-arc connected networks, so that the value $\tilde{V}(Q,k)$ of the smart search game played on such networks is $\mu(1-1/(k+1))$. Recall that a network is 2-arc connected if and only if it cannot be disconnected by the removal of fewer than two arcs. We also show that the bounds given by Theorems 3.3 and 3.8 are equal if Q is a circle.

Suppose a rooted network Q has an MRES P with length $\bar{\mu} = \mu$. Then for any given arc a, let t be the first time P has searched the whole of a. By definition of an expanding search, $P([0,t])$ is connected and contains O, and so is $P([0,t]) - a$. Since P is reversible and minimal, $P^{-1}([0,\mu - t)) = Q - P([0,t])$ is connected and contains O, and by the minimality of P, $Q - P([0,t])$ is disjoint from a. Hence the union $(P([0,t]) - a) \cup (Q - P([0,t])) = Q - a$ is connected, and so Q is 2-arc connected. We have shown that a network is 2-arc connected if it has an MRES of length μ. In fact, in [8], the authors show that the reverse implication also holds, so that we have the following theorem.

THEOREM 3.9. *A network with total measure μ is 2-arc connected if and only if it has an MRES of length μ.*

We can now give the solution of the smart search game $\tilde{\Lambda}(Q,k)$ for 2-arc connected networks.

THEOREM 3.10. *If Q is 2-arc connected, then the value of the smart search game $\tilde{\Lambda}(Q,k)$ is $\tilde{V}(Q,k) = \mu(1-\frac{1}{k+1})$. The k-uniform strategy u_k is optimal for the Hider, and the smart k-uniform MRES $\bar{P}$ is optimal for the Searcher.*

Proof. If the Hider follows u_k, then by Theorem 3.3 he ensures an expected search time of at least $\mu(1-\frac{1}{k+1})$, so $\tilde{V} \geq \mu(1-\frac{1}{k+1})$. If the Searcher follows $\bar{P}$, then by Theorem 3.6 he ensures an expected search time of at most $\bar{\mu}(1-\frac{1}{k+1})$, so $\tilde{V} \leq \bar{\mu}(1-\frac{1}{k+1})$. Since Q is 2-arc connected, $\mu = \bar{\mu}$, by Theorem 3.9, so $\tilde{V} \leq \mu(1-\frac{1}{k+1})$, and we must have equality. □

For a normal Searcher, Theorems 3.8 and 3.9 imply that if Q is 2-arc connected, the value of the game is no more than $\mu(1-1/(2k))$. We sum this up in the theorem below.

THEOREM 3.11. *If Q is 2-arc connected, the value of the normal search game satisfies $V(Q,k) \leq \mu(1-\frac{1}{2k})$.*

This bound is tight in the case that Q is a circle, as we now show. We denote the circle C by the interval $[0,\mu]$, identifying the points 0 and μ, which is the root.

THEOREM 3.12. *The value normal search game $\Lambda(C,k)$ is $V(C,k) = \mu(1-\frac{1}{2k})$. The normal k-uniform MRES P^* is optimal for the Searcher. It is optimal for the Hider to follow the strategy h in which he picks some x uniformly from the interval $[0,\mu/k)$ and hides the objects at the points $\{x, x+\mu/k, x+2\mu/k, \ldots, x+(k-1)\mu/k\}$.*

Proof. By Theorem 3.11, $V(C,k) \leq \mu(1-\frac{1}{2k})$ since C is 2-arc connected, so it remains to be shown that the Hider's strategy h ensures an expected search time no greater than $\mu(1-\frac{1}{2k})$. For any normal Searcher strategy S, $k-1$ of the objects will be found be S by time $\mu(1-1/k)$, and the whole of C will have been searched except for an interval $I = [a, a+\mu/k)$, for some $a \leq \mu(1-1/k)$. The remaining object that has not been found is located uniformly on I, so any search will find it in additional time $\frac{\mu}{2k}$. The total search time is therefore $\mu(1-\frac{1}{k}) + \frac{\mu}{2k} = \mu(1-\frac{1}{2k})$. This completes the proof. □

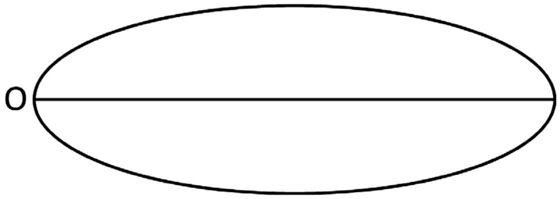

Fig. 3.5. *The three-arc network.*

The bound in Theorem 3.8 is not tight for general 2-arc connected networks, as demonstrated by the 3-arc network Q depicted in Figure 3.5, which consists of three arcs of length 1.

Consider the expanding search game $\Lambda(Q,2)$ with two hiders played on Q with a normal Searcher. Theorem 3.11 gives the bound $V(Q,2) \leq 3(1-1/4) = 9/4$. However, the Searcher can improve on this bound. Consider the Searcher strategy where he picks two arcs at random and uses his optimal strategy for the circle with $k=2$ on this subnetwork, then uses his strategy for $k=1$ on the remaining arc, which can now be regarded as a circle.

Suppose the Hider places his two objects on different arcs. If the Searcher chooses these arcs first (probability 1/3), then by Theorem 3.12, the expected capture time will be $2(1-1/4) = 3/2$. If he chooses a different pair of arcs first (probability 2/3), then again, by Theorem 3.12, the expected capture time will be $2+1(1-1/2) = 5/2$. So the overall expected capture time is $1/3 \cdot 3/2 + 2/3 \cdot 5/2 = 13/6$.

Now suppose the Hider places both objects on the same arc. If one of the first two arcs the Searcher chooses is this one (probability 2/3), the expected capture is $2(1-1/4) = 3/2$. If the Searcher chooses the two other arcs first (probability 1/3), then the expected capture time is no greater than 3. So the overall expected capture time is no greater than $2/3 \cdot 3/2 + 1/3 \cdot 3 = 2 < 13/6$. So $V(Q,2) \leq 13/6 < 9/4$.

It can be shown that $V(Q,2) = 13/6$, as the Hider can ensure expected capture time at least 13/6 by picking two arcs at random and using his optimal strategy for the circle with $k=2$ on this subnetwork. A detailed proof is omitted, but it is sufficient to show that the Searcher has a best response to this Hider strategy which begins by searching only two of the arcs. If the Searcher uses a strategy of this type, then given that he finds the final object before time 2, the expected capture time is 3/2; given that he finds the final object after time 2, the expected capture time is 5/2. Hence the overall expected capture time is precisely $1/3 \cdot 3/2 + 2/3 \cdot 5/2 = 13/6$.

4. Conclusion.

We have defined and solved a natural search problem in which a Searcher wishes to find several balls hidden in boxes with designated search costs. We have shown how the problem can be viewed as a particular case of an extension of the expanding search game for a single object on a network recently introduced in [8]. We have shown that a smart Searcher has no advantage in this game over a normal Searcher but may in the general expanding search game for multiple objects on a network. We have given bounds for the value of this game on an arbitrary network, and we have solved the game for a smart Searcher on a 2-arc connected network and for a normal Searcher on a circle.

The exclusion of overlook probabilities in this model of a box search provides a new class of discrete search problems which are distinct from the type first considered in Matula [20]. The elementary nature of the problem means that it has the potential to be applied to many areas of science.

There are several simple extensions to the games presented here which merit further study. For example, the Searcher may wish to minimize the search cost incurred in finding some $j < k$ of the objects, or he may have a fixed budget, under which he wishes to maximize the number of objects he can find. Alternatively, one might consider a variation of the problem in which a continuous quantity like oil is hidden on a network or in discrete locations.

Acknowledgments. We would like to thank an anonymous referee for many helpful remarks and suggested simplifications to this paper. We also thank Richard Weber for useful discussions.

REFERENCES

[1] S. Alpern, *Hide and Seek Games*, seminar, Institut fur Hohere Studien, Vienna, 1976.

[2] S. Alpern, *The rendezvous search problem*, SIAM J. Control Optim., 33 (1995), pp. 673–683.

[3] S. Alpern, *Search games on trees with asymmetric travel times*, SIAM J. Control Optim., 48 (2010), pp. 5547–5563.

[4] S. Alpern, R. Fokkink, J. Op Den Kelder, and T. Lidbetter, *Disperse or Unite? A Mathematical Model of Coordinated Attack*, Lecture Notes in Comput. Sci. 6442, Springer, New York, 2010, pp. 220–233.

[5] S. Alpern, R. Fokkink, and K. Kikuta, *On Ruckle's conjecture on accumulation games*, SIAM J. Control Optim., 48 (2010), pp. 5073–5083.

[6] S. Alpern, R. Fokkink, T. Lidbetter, and N. S. Clayton, *A search game model of the scatter hoarder's problem*, J. Roy. Soc. Interface, 9 (2012), pp. 869–879.

[7] S. Alpern and S. Gal, *The Theory of Search Games and Rendezvous*, Internat. Ser. Oper. Res. Management Sci. 319, Kluwer Academic, Boston, 2003.

[8] S. Alpern and T. Lidbetter, *Mining coal or finding terrorists: The expanding search paradigm*, Oper. Res. (2013).

[9] E. J. Anderson and R. R. Weber, *The rendezvous problem on discrete locations*, J. Appl. Probab., 28 (1990), pp. 839–851.

[10] D. Assaf and S. Zamir, *Continuous and discrete search for one of many objects*, Oper. Res. Lett., 6 (1987), pp. 205–209.

[11] J. S. Benkoski, M. G. Monticino, and J. R. Weisinger, *A survey of the search theory literature*, Naval Res. Logist., 38 (1991), pp. 469–494.

[12] J. Bram, *A 2-Player N-Region Search Game*, Operations Evaluation Group, Office of Chief Naval Operations, Washington, DC, 1962.

[13] S. Gal, *Search games with mobile and immobile hider*, SIAM J. Control Optim., 17 (1979), pp. 99–122.

[14] S. Gal, *On the optimality of a simple strategy for searching graphs*, Internat. J. Game Theory, 29 (2001), pp. 533–542.

[15] S. Gal, *TPZS applications: (Antagonistic) search games*, in Wiley Encyclopedia of Operations Research and Management Science, J. J. Cochran, ed., Wiley, New York, 2010.

[16] A. Garnaev, *Search Games and Other Applications of Game Theory*, Lecture Notes in Econom. and Math. Systems 485, Springer-Verlag, Berlin, 2000.

[17] J. C. Gittins, K. Glazebrook, and R. R. Weber, *Multi-Armed Bandit Allocation Indices*, 2nd ed., Wiley, New York, 2011.

[18] S. G. Hoggar, *Chromatic polynomials and logarithmic concavity*, J. Combin. Theory Ser. B, 16 (1974), pp. 248–254.

[19] R. Isaacs, *Differential Games*, Wiley, New York, 1965.

[20] D. Matula, *A periodic optimal search*, Amer. Math. Monthly, 71 (1964), pp. 15–21.

[21] T. Nakai, *A preemptive detection game*, J. Inf. Optim. Sci., 11 (1990), pp. 1–15.

[22] M. F. Neuts, *A multistage search game*, J. SIAM, 11 (1963), pp. 502–507.

[23] V. V. Pravosudov and N. S. Clayton, *A test of the adaptive specialization hypothesis: Population differences in caching, memory, and the hippocampus in black-capped chickadees (Poecile atricapilla)*, Behav. Neurosci., 116 (2002), pp. 515–522.

[24] W. Press, *Strong profiling is not mathematically optimal for discovering rare malfeasors*, Proc. Natl. Acad. Sci. USA, 106 (2008), pp. 1716–1719.

[25] D. J. Reyniers, *Coordinating two searchers for an object hidden in an interval*, J. Oper. Res. Soc., 46 (1995), pp. 1386–1392.

[26] D. J. Reyniers, *Coordinated search for an object hidden on a line*, European J. Oper. Res., 95 (1996), pp. 467–685.

[27] W. H. Ruckle, *A Discrete Search Game*, in Stochastic Games and Related Topics, T. E. S. Raghaven, T. S. Ferguson, T. Parthasarathy, and O. J. Vrieze, eds., Kluwer Academic, Boston, 1991, pp. 29–43.

[28] A. Sharlin, *Optimal search for one of many objects hidden in two boxes*, European J. Oper. Res., 32 (1987), pp. 251–259.

[29] L. C. Thomas, *Finding your kids when they are lost*, J. Oper. Res. Soc., 43 (1992), pp. 637–639.

[30] R. R. Weber, *Optimal symmetric rendezvous search on three locations*, Math. Oper. Res., 37 (2012), pp. 111–122.

[31] M. L. Weitzman, *Optimal search for the best alternative*, Econometrica, 47 (1979), pp. 641–654.