Optimal Controlled Flooding Search in a Large Wireless Network

Nicholas B. Chang and Mingyan Liu *
University of Michigan
Electrical Engineering and Computer Science Department
Ann Arbor, USA
{changn,mingyan}@eecs.umich.edu

Abstract

In this paper we consider the problem of searching for a node or an object (i.e., piece of data, file, etc.) in a large wireless network. We consider the class of controlled flooding search strategies where query/search packets are broadcast and propagated in the network until a preset TTL (time-to-live) value carried in the packet expires. Every unsuccessful search attempt results in an increased TTL value (i.e., larger search area) and the same process is repeated. We derive search strategies that minimize the search cost in the worst-case, via a performance measure in the form of the competitive ratio between the average search cost of a strategy and that of an omniscient observer. This ratio is shown in prior work to be lower bounded by 4 among all deterministic search strategies. In this paper we show that by using randomized strategies this ratio is lower bounded by e. We derive an optimal strategy that achieves this lower bound, and discuss its performance under other performance criteria.

1 Introduction

In this paper we consider the problem of searching for a node or an object (e.g., piece of data, file, etc.) in a large wireless network. A prime example is data query in a wireless sensor network, where different sensing data is distributed among a large number of sensor nodes [3]. It has also been extensively used in mobile ad hoc networks, including searching for a destination node by a source node in the route establishment procedure of an ad hoc routing protocol (e.g., [8]), searching for a multicast group by a node looking to join the group (e.g., [12]), and locating one or multiple servers by a node requesting distributed services (e.g., [4]).

*This work is partially supported by NSF grant ANI-0238035 and ONR grant N00014-03-1-0232.

A variety of mechanisms may be used to locate a node in a network. For instance, a centralized directory service, which is periodically updated, can be established from which location information may be obtained. The central directory is constantly updated as the network topology and data content change. Such systems tend to have very short response time, if the directory information is kept afresh. On the other hand, centralized systems often scale poorly as the network increases in size and as location information changes more frequently (either due to topology change as a result of mobility or due to the information content change in the network). The latter necessitates a large amount of information update which can cause significant energy consumption overhead, especially when the queries occur less frequently compared to changes in the network. The central directory may also result in a single point of failure. One can also use the decentralized random walk based search, where the querier sends out a query packet to be forwarded in some random fashion, e.g., random walks or controlled walks such that the propagation of the packet follows a consistent direction, until it hits the search target. For example, [3] proposed random walks initiated by both the querier and the node that has data of potential interest (called advertisement). There have been many results on estimating the search cost and response time using such approaches, see for example [11].

In this paper we focus on a widely used search mechanism known as the TTL-based controlled flooding of query packets. This method is widely used in ad hoc routing protocols [7] as well as wired networks [1]. This is also a decentralized approach in that no central directory of information is maintained. Under this scheme the query/search packet is broadcast and propagated in the network. A preset TTL (time-to-live) value is carried in the packet and every time the packet is relayed the TTL value is decremented. This continues until TTL reaches zero and the propagation stops. Therefore the extent/area of the search is controlled by the TTL value. If the target is located within this area, it will reply with the queried information. Otherwise, the origin of the search will eventually time out and initiate another round of search covering a bigger area using a larger TTL value. This continues until either the object is found or the querier gives up. Consequently the performance of a search strategy is determined by the sequence of TTL values used.

Compared to random walk based approaches, controlled flooding search is much easier to implement, and likely results in shorter response times on average. On the other hand, it may incur more energy consumption (in terms of number of transmissions and receptions needed) in the network if not used properly.

Our primary goal is to derive controlled flooding search strategies, i.e., sequences of TTL values, that minimize the cost of such searches in terms of energy consumption (i.e., the amount of packet transmission/reception) [1]. We will also limit our analysis to the case of searching for a single target, which is assumed to exist in the network. For the rest of our discussion we will use the term *object* to indicate the target of a search, be it a node, a piece of data or a file. We will measure the position of an object by its distance to the source originating the searching. We will use the term *location* of an object to indicate both the actual position of the search target within the network and the minimum TTL value required to locate this object. The terms *search strategy* or simply *strategy* will take on a more limited meaning within the context of controlled flooding search and refer to a TTL sequence.

When the probability distribution of the location of the object is known *a priori*, search strategies that minimize the expected search cost can be obtained via a dynamic programming formulation [6]. The necessary and sufficient conditions were also derived in [6] for two very commonly used search strategies to be optimal. When the distribution of the object location is not known *a priori*, one may evaluate the effectiveness of a strategy by its worst case performance. In [1] such a criterion, in the form of the competitive ratio (or worst-case cost ratio) between the expected cost of a given strategy and that of an omniscient observer, was used and it was shown under a linear cost model (to be precisely defined in the next section) that the best worst-case search strategy among all *fixed strategies* is the California Split Search algorithm, which achieves a competitive ratio of 4 (also the lower bound on all fixed strategies). In [6] it was shown that to minimize this ratio, the best strategies are *randomized* strategies that consist of sequence of random variables, i.e., successive TTL values are drawn from certain probability distributions rather than deterministic values. In particular, it was shown that given a deterministic TTL sequence, there exists a randomized version that has a lower worst-case expected search cost. [6] introduced a class of *uniformly randomized strategies* and showed that within this class the best strategy achieves a competitive ratio of approximately 2.9142. In this paper we show that for a much more general class of cost models, the best worst-case strategy among all fixed and random strategies achieves a worst-case cost ratio of e. We derive an optimal randomized strategy that attains this ratio and discuss how it can be adjusted to account for alternative performance criteria.

The rest of the paper is organized as follows. Sections 2 and 3 present the network model and the performance objectives under consideration. In Section 4 we derive the optimal strategy among all random and non-random strategies. We examine a few alternative performance measures in Section 5, discuss some practical implications in Section 6, and conclude the paper in Section 7.

2 Network model

We will assume that the timeout values are perfectly set such that when the timer expires for a query with TTL value k, that query has reached all nodes k hops away. We denote by L the minimum TTL value required to search every node within the network, and will also refer to L as the *dimension* or *size* of the network.

A search strategy $\mathbf{u}$ is a TTL sequence of certain length N, $\mathbf{u} = [u_1, u_2, \cdots, u_N]$. It can be either fixed/deterministic where $u_i, i = 1, \cdots, N$, are deterministic values, or random where u_i are drawn from probability distributions. For a fixed strategy we assume that $\mathbf{u}$ is an increasing sequence. In practice, it is natural to only consider integer-valued policies. However, considering real-valued sequences can often reveal fundamental properties that are helpful in deriving optimal integer-valued strategies. We therefore also consider continuous (real-valued) strategies, denoted by $\mathbf{v}$, where $\mathbf{v} = [v_1, v_2, \cdots, v_N]$, and v_i is either a fixed or continuous random variable that takes any real value on $[1, \infty)$, for $1 \leq i \leq L$.

A strategy is *admissible* if it locates any object of finite location with probability 1. For a fixed strategy this implies $u_N = L$, and for a random strategy, this implies $Pr(u_i = L) = 1$ for some $1 \leq i \leq N$. In the asymptotic case as $L \to \infty$, a strategy $\mathbf{u}$ is admissible if $\forall\ x \geq 1,\ \exists\ n \in \mathbb{Z}^+$ s.t. $Pr\left(u_n \geq x\right) = 1$.

We let V denote the set of all real-valued admissible strategies (random or fixed). V^d denotes the set of all admissible real-valued deterministic strategies. U denotes the set of all integer-valued admissible strategies (random or fixed). Finally, U^d denotes the set of all admissible integer-valued deterministic strategies. Note that it is always true that $U^d \subset U \subset V$, and similarly $U^d \subset V^d \subset V$.

In a practical system, a variety of techniques may be

[1] We will not explicitly consider the response time of a search strategy in this paper, as within the class of controlled flooding search the fastest search is to flood the entire network.

used to reduce the number of query packets flowing in the network and to alleviate the *broadcast storm* problem [10]. In our analysis we will assume that a search with a TTL value of k will reach all neighbors that are k hops away from the originating node, and that the cost associated with this search is a function of k, denoted by $C(k)$. This cost may include the total number of transmissions, receptions, etc. Thus $C(k)$ is the ultimate abstraction of the nature of the underlying network and the specific broadcast schemes used.

For real-valued sequences, we require that the cost function $C(v)$ be defined for all $v \in [1, \infty)$, while for integer-valued sequences we only require that the cost function be defined for positive integers. When the cost function is invertible, we write $C^{-1}(\cdot)$ to denote its inverse. We will adopt the natural assumption that $C(v_1) > C(v_2)$ if $v_1 > v_2$. We also denote by $\mathbb{C}$ the class of cost functions $C : [1, \infty) \to [C(1), \infty)$, that are increasing, differentiable, and have the property $\lim_{v\to\infty} C(v) = \infty$.

Two example cost functions are the linear cost and quadratic cost, defined as $C(v) = \alpha v$ and $C(v) = \alpha v^2$, respectively, for some constant $\alpha > 0$. The first is a good model in a network where the number of transmissions incurred by the search query is proportional to the TTL value used, e.g., in a linear network with constant node density. The latter is a more reasonable model for a two-dimensional network, as the number of nodes reached (as well as the number of transmissions) in v hops is on the order of v^2 [1].

We will use X to denote the minimum TTL value required to locate the object. We will also loosely refer to X as the object "location". When considering discrete strategies $\mathbf{u} \in U$, we require X to be an integer-valued random variable taking values between 1 and L such that $Pr(X \in \{1, 2, \cdots, L-1, L\}) = 1$. When considering continuous strategies $\mathbf{v} \in V$, we relax the integer restriction on X, and allow the location to take any real value in the interval $[1, L]$. In both cases, we denote the cumulative distribution of X by $F(x)$, where $F(x) = Pr(X \le x)$. Similarly, the tail distribution of X is denoted by $\bar{F}(x) = 1-F(x) = Pr(X > x)$. Note that $F(L) = 1$ and $\bar{F}(L) = 0$ for any X.

3 Problem formulation and preliminaries

We adopt the following worst-case performance measure (a generalization of the one used in [1]):

$$\rho^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} , \qquad (1)$$

where $J_X^{\mathbf{u}}$ denotes the expected search cost of using strategy $\mathbf{u}$ for object location X; $E[C(X)]$ is the expected search cost of an ideal omniscient observer who knows precisely the location (i.e., realization of X). The ratio between these two terms for a given X will be referred to as the (expected) *cost ratio*. $\{p_X(x)\}$ denotes the set of all probability mass functions of X such that $E[C(X)] < \infty$. We will only consider the case where the random vector $\mathbf{u}$ and X are mutually independent. Also note that if $\mathbf{u}$ is deterministic then $J_X^{\mathbf{u}}$ is a single expectation with respect to X, whereas if $\mathbf{u}$ is random then $J_X^{\mathbf{u}}$ is the average over both X and $\mathbf{u}$. The worst-case cost ratio $\rho^{\mathbf{u}}$ can also be viewed as the *competitive ratio* with respect to an *oblivious adversary* [2] who knows the search strategy $\mathbf{u}$. We will use the two terms interchangeably.

It should be mentioned that the quantity $\rho^{\mathbf{u}}$ has slightly different meanings for deterministic and randomized strategies. When $\mathbf{u}$ is a fixed sequence $J_X^{\mathbf{u}}$ is a single expectation with respect to X as noted before. In this case, the search cost of using $\mathbf{u}$ is always within a factor $\rho^{\mathbf{u}}$ of the omniscient observer cost for any given location. On the other hand, when $\mathbf{u}$ is random, $\rho^{\mathbf{u}}$ only provides an upper bound on the average search cost but does not necessarily upper bound any particular realization of this cost, as $J_X^{\mathbf{u}}$ is a double expectation with respect to both the strategy and the location. In this case, it is the *expected* search cost of $\mathbf{u}$ that is always within $\rho^{\mathbf{u}}$ of the cost of an omniscient observer. For some realizations of $\mathbf{u}$ and X, the cost ratio may be higher than $\rho^{\mathbf{u}}$. In Section 5, we will present other performance measures in order to account for these differences.

The corresponding objective is to find search strategies that minimize this ratio, with the best worst-case discrete strategy denoted by $\mathbf{u}^*$:

$$\rho^* = \inf_{\mathbf{u}\in U} \rho^{\mathbf{u}} = \inf_{\mathbf{u}\in U} \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} . \qquad (2)$$

For any continuous strategy, $\mathbf{v} \in V$, the worst-case cost ratio is similarly defined as in (1):

$$\rho^{\mathbf{v}} = \sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} , \qquad (3)$$

where $\{f_X(x)\}$ denotes the set of all probability density functions for X such that $E[C(X)] < \infty$. The best worst-case continuous strategy $\mathbf{v}^* \in V$ is similarly defined as in (2) with $\{f_X(x)\}$ replacing $\{p_X(x)\}$.

The following lemmas are critical in our subsequent analysis.

Lemma 1. *For any search strategy* $\mathbf{v} \in V$, *we have*

$$\sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} = \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)} , \qquad (4)$$

where $J_X^{\mathbf{v}}$ *is the expected search cost using TTL sequence* $\mathbf{v}$ *when object location* X *has pdf* $f_X(x)$*, and* $J_x^{\mathbf{v}}$ *is the expected search cost using TTL sequence* $\mathbf{v}$ *when object location density is* $f_X(x') = \delta(x' - x)$*, i.e., a single fixed point.*

Proof. We begin by noting that for every $x \in [1, \infty)$, there corresponds a singleton probability density $f_X(x') = \delta(x' - x)$, such that $E[C(X)] = C(x)$ and $J_X^{\mathbf{v}} = J_x^{\mathbf{v}}$. We thus have the following inequality

$$\sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} \geq \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)} , \tag{5}$$

since the left-hand side is a supremum over a larger set.

On the other hand, setting $A = \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)}$ we have $\frac{J_x^{\mathbf{v}}}{C(x)} \leq A$ for all $x \in [1, \infty)$. Thus $J_x^{\mathbf{v}} \leq AC(x)$. Then for any random variable X denoting object location, we can use this inequality along with the independence between $\mathbf{v}$ and X to obtain:

$$\begin{aligned}\frac{J_X^{\mathbf{v}}}{E[C(X)]} &= \frac{\int_{[1,\infty)} J_x^{\mathbf{v}} f_X(x)\, dx}{\int_{[1,\infty)} C(y) f_X(x)\, dx} \\ &\leq \frac{\int_{[1,\infty)} AC(x) f_X(x)\, dx}{\int_{[1,\infty)} C(x) f_X(x)\, dx} = A\,. \end{aligned} \tag{6}$$

Equation (6) implies that $\frac{J_X^{\mathbf{v}}}{E[C(X)]} \leq A = \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)}$. Since this inequality holds for all possible random variables X, we have:

$$\sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} \leq \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)} . \tag{7}$$

Inequalities (5) and (7) collectively imply the equality in Lemma 1, thereby completing the proof. □

Because $U \subset V$, the following result can be proven in a very similar fashion to Lemma 1. Alternatively, one can find the proof in [6].

Lemma 2. *For any search strategy $\mathbf{u} \in U$, we have*

$$\rho^{\mathbf{u}} = \sup_{\{p_X(x)\}} \frac{J_X^{\mathbf{u}}}{E[C(X)]} = \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)}, \tag{8}$$

where $J_x^{\mathbf{u}}$ denotes the expected search cost using TTL sequence $\mathbf{u}$ when $Pr(X = x) = 1$, and $\mathbb{Z}^+$ denotes the set of natural numbers.

In words, these lemmas imply that for any TTL sequence, the worst case scenario is when the object location is a constant with a singleton probability distribution, subsequently referred to as a *point*. This result allows us to limit our attention to singleton-valued X and equivalently redefine the minimum worst-case cost ratio ρ^* in equation (2) as

$$\rho^* = \inf_{\mathbf{u}\in U} \rho^{\mathbf{u}} = \inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)}, \tag{9}$$

and similarly for continuous strategies.

It has been shown in [1] that under a linear cost function $C(u) = \alpha \cdot u$ for some constant α, and as the network size increases, the minimum worst-case cost ratio over all deterministic integer-valued sequences is 4, achieved by the California Split Search $\bar{\mathbf{u}} = \{2^{i-1} : i \in \mathbb{Z}^+\} = [1, 2, 4, 8, ...]$. In the next section we derive randomized strategies that are optimal among *all* admissible strategies. Whereas [1] and [6] derive strategies under linear cost functions, our optimal strategy achieves a much smaller worst-case cost ratio, e, for any cost function $C(\cdot) \in \mathbb{C}$.

4 Optimal worst-case strategies

In this section, we derive asymptotically optimal continuous and discrete strategies in the limit as the network dimension $L \to \infty$. Consequently we will consider TTL sequences of infinite length that are admissible as outlined earlier. The asymptotic case is studied as we are particularly interested in the performance of flooding search in a large network. In addition, it is difficult if at all possible to obtain a general strategy that is optimal for all finite-dimension networks because the optimal TTL sequence often depends on the value of L. In this sense, an asymptotically optimal strategy may provide much more insight into the intrinsic structure of the problem. We will see that asymptotically optimal TTL sequences can also perform very well in a network of arbitrary finite dimension.

In what follows we will first derive a tight lower bound on the worst-case cost ratio for continuous strategies. We then introduce a particular randomized continuous strategy that achieves the lower bound, therefore proving that this strategy is optimal in the worst-case. We then repeat the process for the discrete case.

In deriving a tight lower bound on the worst-case cost ratio, we first use Yao's minimax principle [2] and Lemma 1 to obtain the following inequality:

Lemma 3.

$$\sup_{\{f_X(x)\}} \inf_{\mathbf{v}\in V^d} \frac{J_X^{\mathbf{v}}}{E[C(X)]} \leq \inf_{\mathbf{v}\in V} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)}. \tag{10}$$

Proof. First note that for any given object probability distribution, there exists an optimal strategy that is deterministic. Hence the following holds:

$$\sup_{\{f_X(x)\}} \inf_{\mathbf{v}\in V^d} \frac{J_X^{\mathbf{v}}}{E[C(X)]} = \sup_{\{f_X(x)\}} \inf_{\mathbf{v}\in V} \frac{J_X^{\mathbf{v}}}{E[C(X)]} . \tag{11}$$

We also have the following in interchanging the supremum and infimum, see for example [9]:

$$\sup_{\{f_X(x)\}} \inf_{\mathbf{v}\in V} \frac{J_X^{\mathbf{v}}}{E[C(X)]} \leq \inf_{\mathbf{v}\in V} \sup_{\{f_X(x)\}} \frac{J_X^{\mathbf{v}}}{E[C(X)]} . \tag{12}$$

Finally, applying Lemma 1 to the right-hand side of (12) and combining this inequality with (11) establishes (10). □

Using this lemma, we note that any lower bound can be found by first selecting a location distribution $f_X(x)$ and deriving the optimal deterministic strategy that minimizes the cost ratio under this distribution. We will assume that the cost function $C(x) \in \mathbb{C}$.

Consider an object location distribution given by $\bar{F}(x) = Pr(X > x) = \left(\frac{C(x)}{C(1)}\right)^{-\alpha}$ for all $x \geq 1$ and some constant $\alpha > 1$. A special case of this distribution where cost $C(\cdot)$ is linear, also known as the Zipf distribution, was studied in [1] for which the optimal deterministic strategy was computed. Here we will follow a similar method (generalized to any cost function in $\mathbb{C}$) to derive the class of optimal strategies. For any deterministic TTL sequence $\mathbf{v} = [v_1, v_2, ...]$, the corresponding expected search cost is given by the following expression, where $v_0 = 1$ is assumed for simplicity of notation:

$$J_X^{\mathbf{v}} = \sum_{j=1}^{\infty} C(v_j)\bar{F}(v_{j-1}) = \sum_{j=1}^{\infty} C(v_j)\left(\frac{C(v_{j-1})}{C(1)}\right)^{-\alpha} . \tag{13}$$

Taking the partial derivative of (13) with respect to v_j and then setting this equal to 0 gives a necessary condition for any optimal strategy, for all $j \geq 1$. From this condition, it can be shown that for a given fixed v_1, the optimal strategy is to recursively choose v_j that satisfy the following equation for all $j \geq 1$:

$$C(v_{j+1}) = \frac{C(v_j)}{\alpha}\left(\frac{C(v_j)}{C(v_{j-1})}\right)^{\alpha} . \tag{14}$$

From this, it can be shown [5] that the optimal strategy must satisfy $J_X^{\mathbf{v}}\frac{\alpha-1}{\alpha} = C(v_1)$. On the other hand, the mean of the object location cost can be calculated by integrating the tail distribution of the random variable $C(X)$, giving $E\left[C(X)\right] = \frac{\alpha}{\alpha-1}C(1)$. Combining these two results gives:

$$\frac{J_X^{\mathbf{v}}}{E\left[C\left(X\right)\right]} = J_X^{\mathbf{v}}\frac{(\alpha-1)}{\alpha C(1)} = \frac{C(v_1)}{C(1)} . \tag{15}$$

This result implies that for a given α, the sequence that generates the smallest cost ratio will follow recursion (14) and use the smallest possible value of v_1. However, not all values of v_1 lead to an increasing sequence $\mathbf{v}$, which is obviously a requirement for an optimal strategy. It can be shown [5] that $\mathbf{v}$ is an increasing sequence if and only if $\frac{C(v_1)}{C(1)} > \alpha^{\frac{1}{\alpha-1}}$. Therefore, for a given $\alpha > 1$, the minimum cost ratio of any continuous strategy is lower-bounded by $\alpha^{\frac{1}{\alpha-1}}$.

Using recursion (14) a TTL sequence is completely defined by the selection of v_1. Therefore we can come arbitrarily close to the value $\alpha^{\frac{1}{\alpha-1}}$ by using a TTL sequence defined by some v_1 such that $\frac{C(v_1)}{C(1)}$ is arbitrarily close to $\alpha^{\frac{1}{\alpha-1}}$.

To establish a tight lower bound in Lemma 3, we need to find the value of α with the highest minimum cost ratio. It can be seen that as α approaches 1 from above, the minimum cost ratio increases. Thus the maximum value of this can be calculated by taking the limit $\lim_{\alpha\to1^+}\alpha^{\frac{1}{\alpha-1}} = e$. Hence from Lemma 3 we have the following:

Lemma 4. *For any $C(x) \in \mathbb{C}$, the worst-case cost ratio of any continuous strategy is lower-bounded by e, i.e.,*

$$\inf_{\mathbf{v}\in V}\sup_{x\in[1,\infty)}\frac{J_x^{\mathbf{v}}}{C(x)} \geq e. \tag{16}$$

This result implies that if we can obtain a TTL sequence whose worst-case ratio is e, then it is optimal. We consider the class of strategies given by the following definition:

Definition 1. *Assume $C(x) \in \mathbb{C}$. $\mathbf{v}[r, F_{v_1}(x)]$ denotes a jointly defined sequence $\mathbf{v} = [v_1, v_2, ...]$ with a configurable parameter r generated as follows:*

(J.1) *The first TTL v_1 is a continuous random variable taking values in the interval $\left[1, C^{-1}(r\cdot C(1))\right)$, with its cdf given by a nondecreasing (right-continuous) function $F_{v_1}(x) = Pr(v_1 \leq x)$.*

(J.2) *The k-th TTL v_k is defined by the equation $v_k = C^{-1}\left(r^{k-1}C(v_1)\right)$ for all positive integers k.*

From (J.1) and (J.2), it can be seen that r and $F_{v_1}(x)$ uniquely define the TTL strategy.

For this family of strategies, the space over which the supremum is taken in order to calculate the worst-case cost ratio can be reduced. In particular, we have the following:

Lemma 5. *Consider any strategy $\mathbf{v}[r, F_{v_1}(x)]$ constructed using steps* (J.1) *and* (J.2) *in Definition 1. Assume $C(x) \in \mathbb{C}$. Let $\bar{F}_{v_1}(y) = 1 - F_{v_1}(y)$. Then the worst-case cost ratio $\sup_{x\in[1,\infty)}\frac{J_x^{\mathbf{v}}}{C(x)}$ is given by:*

$$\sup_{1\leq z<r}\left\{\frac{r}{r-1}\frac{h(r)+(r-1)h(z)}{zC(1)} - r\frac{h'(z)}{C(1)}\right\} , \tag{17}$$

where $h(z)$ is defined as follows for $1 \leq z < r$:

$$h(z) = C(1) + \int_{C(1)}^{z\cdot C(1)}\bar{F}_{v_1}(C^{-1}(y))\,dy , \tag{18}$$

and $h'(z)$ denotes the derivative of h with respect to z.

Proof of this lemma can be found in [5]. Using this lemma, for a given r the corresponding distribution $F_{v_1}(x)$ that minimizes worst-case cost ratio can be determined by trying to produce a smooth cost ratio curve, e.g. one in which (17) has a derivative of 0 with respect to z. The reason for this will become clearer in Section 5. These steps lead to the following result.

Theorem 1. *Assume $C(x) \in \mathbb{C}$. Within the class of continuous (real-valued) TTL sequences V, the smallest obtainable worst-case cost ratio is e, i.e.,*

$$\inf_{\mathbf{v}\in V} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)} = e.$$

Moreover, this worst-case ratio is obtained by the strategy $\mathbf{v}^*[e, \ln\frac{C(x)}{C(1)}]$ as defined in Definition 1.

To prove the theorem, first apply Lemma 5 to show that the worst-case cost ratio of $\mathbf{v}^*[e, \ln\frac{C(x)}{C(1)}]$ is equal to e. Theorem 1 then follows from Lemma 4.

As an example, when the cost is linear, i.e. $C(x) = \alpha x$ for all x, the optimal strategy $\mathbf{v}^* = [v_1^*, v_2^*, ...]$ is defined as follows. The first TTL value is a random variable v_1^* with cdf $F_{v_1^*}(z) = \ln z$ for $1 \le z < e$. Successive TTL values are defined as $v_k^* = e^{k-1}v_1^*$.

In addition, it follows from Lemma 5 that a strategy $\mathbf{v}[r, F_{v_1}(x)]$ given in Definition 1, with $F_{v_1}(x) = \frac{1}{\ln r}\ln\frac{C(x)}{C(1)}$, has a worst-case cost ratio of $\frac{r}{\ln r}$. We will consider this family of strategies later when discussing other performance criteria.

Using the above optimal continuous strategy, we next derive an optimal discrete strategy.

Lemma 6. *For $C(x) \in \mathbb{C}$ we have*

$$\inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)} \ge \inf_{\mathbf{v}\in V} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)} = e. \quad (19)$$

That is, the minimum worst-case cost ratio over all integer-valued strategies is lower-bounded by the minimum worst-case cost ratio over all real-valued strategies.

Proof. Consider any discrete strategy $\mathbf{u} \in U$. It is also true that $\mathbf{u} \in V$. Note that for any $0 < \epsilon < 1$ and positive integer x, we have $J_{x+\epsilon}^{\mathbf{u}} = J_x^{\mathbf{u}}$. Hence:

$$\frac{J_{x+\epsilon}^{\mathbf{u}}}{C(x+\epsilon)} < \frac{J_x^{\mathbf{u}}}{C(x)} \ ,$$

since the cost function is strictly increasing. Thus the worst-case cost ratio occurs at some positive integer x. This means that the following is true:

$$\inf_{\mathbf{u}\in U} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{u}}}{C(x)} = \inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)} \ .$$

Since $U \subseteq V$, we have the following:

$$\inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)} = \inf_{\mathbf{u}\in U} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{u}}}{C(x)} \ge \inf_{\mathbf{v}\in V} \sup_{x\in[1,\infty)} \frac{J_x^{\mathbf{v}}}{C(x)},$$

where the right inequality holds because the infimum is over a larger set, thus completing the proof. ☐

This result says that if we can find a discrete strategy whose worst-case cost ratio is e, then it is optimal among all admissible discrete strategies.

Theorem 2. *Consider the cost function $C(x) \in \mathbb{C}$. Within the class of discrete TTL sequences, the best worst-case cost ratio is e, i.e.,*

$$\inf_{\mathbf{u}\in U} \sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}}}{C(x)} = e.$$

Moreover, this worst-case ratio is obtainable by the strategy $\mathbf{u}^*$, which is constructed as follows. Construct a real-valued TTL sequence $\mathbf{v}^*$ by using the strategy $\mathbf{v}^*[e, \ln\frac{C(x)}{C(1)}]$ described in Theorem 1. Then set $u_k^* = \lfloor v_k^* \rfloor$ for all k to obtain the discrete strategy $\mathbf{u}^* = [u_1^*, u_2^*, ...]$.

Proof. Consider the strategies $\mathbf{u}^*$ and $\mathbf{v}^*$ as described in the theorem. Lemma 6 implies that $\sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}^*}}{C(x)} \ge e$. Therefore, to complete the proof we need to show that the worst-case cost ratio of $\mathbf{u}^*$ is less than or equal to e.

For any positive integer k, u_k^* takes integer values between $\lfloor C^{-1}(e^{k-1}C(1)) \rfloor$ and $\lfloor C^{-1}(e^k C(1)) \rfloor$. In addition, $\lfloor C^{-1}(e^k C(1)) \rfloor$ is a nondecreasing sequence with respect to integer values of k, and approaches ∞ as k approaches ∞. Fix the object location as a positive integer x, and choose the smallest integer k such that $x \le \lfloor C^{-1}(e^k C(1)) \rfloor$. Note that $E[C(u_k^*)] = E[C(\lfloor v_k^* \rfloor)] \le E[C(v_k^*)]$ for all integers k. Since x is a positive integer, we have $Pr(u_k^* < x) = Pr(\lfloor v_k^* \rfloor < x) = Pr(v_k^* < x)$. Hence we have the following for this x:

$$J_x^{\mathbf{u}^*} = \sum_{j=1}^{k} E[C(u_j^*)] + Pr(u_k^* < x)\ E[C(u_{k+1}^*) \,|\, u_k^* < x]$$

$$\le \sum_{j=1}^{k} E[C(v_j^*)] + Pr(v_k^* < x)\, E[C(v_{k+1}^*) \,|\, v_k^* < x]$$

$$= J_x^{\mathbf{v}^*} \le eC(x)\ ,$$

where the last inequality holds because the worst-case cost ratio for $\mathbf{v}^*$ is e as proven in Theorem 1. Since this result holds for all integers x, we have $\sup_{x\in\mathbb{Z}^+} \frac{J_x^{\mathbf{u}^*}}{C(x)} \le e$. This implies that the worst-case cost ratio is exactly e. Finally, from Lemma 6 it follows that the minimum worst-case cost ratio of all discrete sequences is e, which means that $\mathbf{u}^*$ is optimal. ☐

As an example, consider when the cost is given by the function $C(x) = \alpha(x-1)^2 + \beta$ for some positive constants α and β. Note that $C^{-1}(y) = \sqrt{(y-\beta)/\alpha} + 1$ for $y \ge \beta$. The optimal continuous strategy $\mathbf{v}^*$ is constructed as follows. From Theorem 1, the first TTL value v_1^* is a continuous random variable taking values in the interval $\left[1, 1+\sqrt{\beta(e-1)/\alpha}\right)$, and with cdf given by $F_{v_1^*}(x) =$

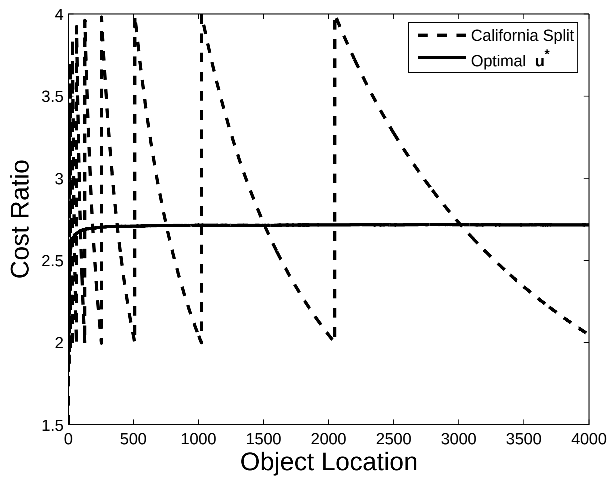

Figure 1. Cost ratio as a function of object location for the optimal discrete sequence $\mathbf{u}^*$ described in Theorem 2, and California Split Search defined by $u_k = 2^{k-1}$ for all k. Cost is assumed to be linear

$\ln\left(\alpha(x-1)^2/\beta+1\right)$. Successive TTL values are defined as $v_k^* = 1 + \sqrt{e^{k-1}(v_1^*-1)^2 + \beta(e^{k-1}-1)/\alpha}$. Then from Theorem 2, the optimal discrete strategy $\mathbf{u}^*$ is derived by setting $u_k^* = \lfloor v_k^* \rfloor$ for all k.

5 Performance comparison

In Figure 1 we compare the cost ratio of the optimal discrete strategy given by Theorem 2 to that of the non-random TTL sequence given by the California Split Search $u_k = 2^{k-1}$ for all k under the linear cost function $C(k) = \alpha k$ for $\alpha > 0$. We see that the cost ratio oscillates for the fixed TTL sequence while randomization essentially has the *averaging* effect that "smooths out" the cost ratio across neighboring locations/points. In fact the curve of the optimal continuous strategy does not have local minima or maxima. One may view this as the built-in *robustness* of a randomized policy for the underlying criterion of worst-case performance. Also note that the worst-case cost ratio e is reached asymptotically from below as $L \to \infty$, and hence the cost ratio at any finite object location is less than the worst-case cost ratio.

The performance measure we have been using is the worst-case cost ratio with respect to an oblivious adversary, who knows the strategy but not every realization of the strategy. However, the same randomized strategy can result in different realizations. This leads us to consider the competitive ratio with respect to an *adaptive offline adversary* [2] who knows the *realization* of the real-valued strategy $\mathbf{v}$ for every search. Let the *worst-realization cost ratio* $\Gamma_X^{\mathbf{v}}$ denote the maximum (over all realizations of strategy $\mathbf{v}$) cost ratio for strategy $\mathbf{v}$ when the object location is a random variable X. Specifically,

$$\Gamma_X^{\mathbf{v}} = \sup_{\tilde{\mathbf{v}} \in \Upsilon^{\mathbf{v}}} \frac{J_X^{\tilde{\mathbf{v}}}}{E\left[C(X)\right]}\,, \tag{20}$$

where $\Upsilon^{\mathbf{v}}$ denotes the set of all possible realizations of strategy $\mathbf{v}$. Let the *worst-case worst realization cost ratio* $\Gamma^{\mathbf{v}}$ denote the maximum of $\Gamma_X^{\mathbf{v}}$ over all possible object locations. Then the performance of a search strategy against an adaptive offline adversary can be measured by the following competitive ratio (worst-case, worst-realization):

$$\Gamma^{\mathbf{v}} = \sup_{\{f_X(x)\}} \Gamma_X^{\mathbf{v}} = \sup_{x\in[1,\infty)} \Gamma_x^{\mathbf{v}}\,, \tag{21}$$

where the second equality can be shown in a manner similar to the proof of Lemma 1. To distinguish, we will refer to $\rho^{\mathbf{v}}$ as the *worst-case average cost ratio*.

As discussed in [2], the minimum of $\Gamma^{\mathbf{v}}$ over all strategies is the same as the minimum worst-case average cost ratio of all deterministic strategies, which can be shown to be 4 under $C(x) \in \mathbb{C}$ [2].

Theorem 3. *Consider a real-valued randomized strategy $\mathbf{v}[r, F_{v_1}(x)]$ that is constructed as given by Definition 1. Then we have $\Gamma^{\mathbf{v}} \leq \frac{r^2}{r-1}$.*

Proof of this result can be found in [5]. The inequality in this theorem becomes an equality when the pdf of v_1 is strictly positive in the interval $\left[C^{-1}\left(\frac{C(x)}{r^{k-1}}\right) - \epsilon, C^{-1}\left(\frac{C(x)}{r^{k-1}}\right)\right)$, for some $\epsilon > 0$. This is true when $F_{v_1}(x) = \frac{1}{\ln r}\ln\frac{C(x)}{C(1)}$, and hence strategies with this family of cdf have a worst-case worst-realization cost ratio of $r^2/(r-1)$, which can be adjusted by selecting the appropriate value of r.

Similar quantities can be defined for best-realization. These results for this family of continuous strategies are depicted in Figure 2 as a function of r. As can be seen, one can appropriately select the value of r depending on whether the goal is to minimize worst-case average cost ratio, worst-case worst-realization cost ratio, etc. In particular, we note that by using $r = 2$, we can obtain a worst-case worst-realization cost ratio of 4, with a worst-case average cost ratio of approximately 2.8854. Therefore this particular strategy strictly outperforms the deterministic California Split Rule. The performance of the optimal continuous strategy, i.e. when $r = e$ as stated in Theorem 1, is depicted in Figure 3 as a function of object location cost.

[2] This can be shown in a similar manner to that used in [1] for discrete strategies under linear cost. In particular, in [5] we establish an equivalency between linear and general cost functions, which can used to show that 4 is minimum worst-case cost ratio among deterministic strategies under any $C(x) \in \mathbb{C}$

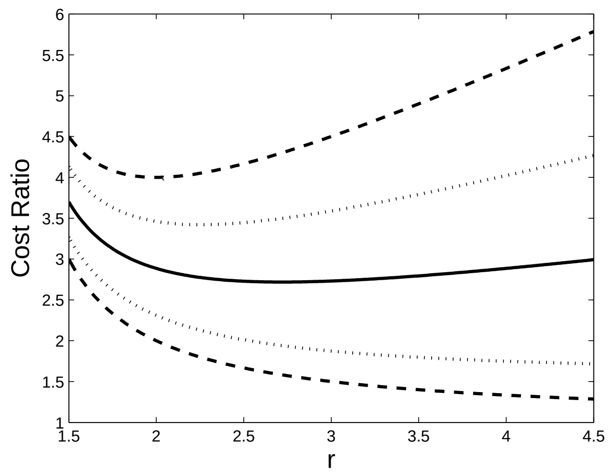

Figure 2. Performance of strategies given by Definition 1 with cdf $F_{v_1}(x) = \frac{1}{\ln r} \ln \frac{C(x)}{C(1)}$, as a function of r. Worst-case average cost ratio (solid), worst and best realization cost ratio for location $x \to \infty$ (dashed), and worst-case average cost ratio $+/-$ standard deviation (dotted) are shown.

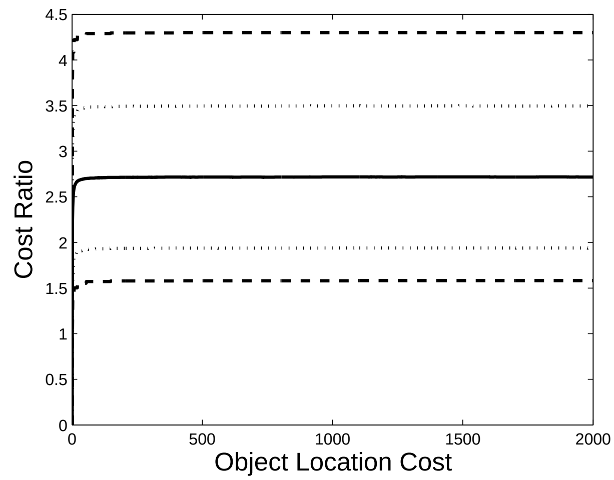

Figure 3. Performance of optimal continuous strategy (Theorem 1) as a function of object location cost. Worst and best realization cost ratio (top and bottom dashed), the average cost ratio (solid), average cost ratio $+/-$ one standard deviation (top and bottom dotted) are shown.

Similar analysis can be carried out for discrete strategies, although in this case the calculations are much more complicated and do not provide any more insight. We therefore do not present the numerical calculations here, but note that the performance is very similar to its continuous version.

6 Discussion

In this paper we have introduced a class of optimal randomized strategies. The derived optimal continuous and discrete randomized strategies rely on the knowledge of the functional form of the search cost $C(\cdot)$. Specifically, construction of the optimal strategy depends on the ability to define and invert a cost function that is defined for all $x \in [1, \infty)$. While conceptually and fundamentally appealing, this construction may pose a problem in a practical setting. Note that the physical meaning of search costs only exists over integer values, while continuous cost functions are introduced as a mathematical tool. If the search cost is only known for integer TTL values, then in order to obtain the optimal discrete search strategy given in Theorem 2, we would need to interpolate and create an increasing, differentiable, and continuous cost function defined over the positive real line.

Such a process is not always easy to carry through. In this case certain approximation may be used. Alternatively we could also try to develop simpler randomized strategies that are sub-optimal with respect to our performance measure but still outperform deterministic strategies and that are much easier to derive and implement than those introduced in Section 4. Such strategies, known as *uniformly randomized*, were introduced in [6] for linear cost. These strategies can be adjusted for more general cost functions as described in [5].

The worst-case cost ratio we have been using so far is in general a conservative/pessimistic performance measure. As mentioned earlier, if the probability distribution of the location of the object is known *a priori*, then we can derive the optimal strategy that achieves the lowest average cost for the given object distribution, using a dynamic programming formulation [6]. On the other hand, the optimal average-cost strategy can potentially be highly sensitive to small disturbances to our knowledge about the object location distribution, while worst-case strategies may be more robust.

We compare the two under the following example scenarios. Consider a network of finite dimension L and the linear cost function $C(k) = k$. We examine what happens when there are errors in our estimate of the location distribution. Consider when the object location has probability mass function $P(X = x) = \beta x^{\alpha}$ for all $1 \leq x \leq L$, where the constant α defines the distribution and β is a normalizing constant. Note that $\alpha = 0$ corresponds to uniform

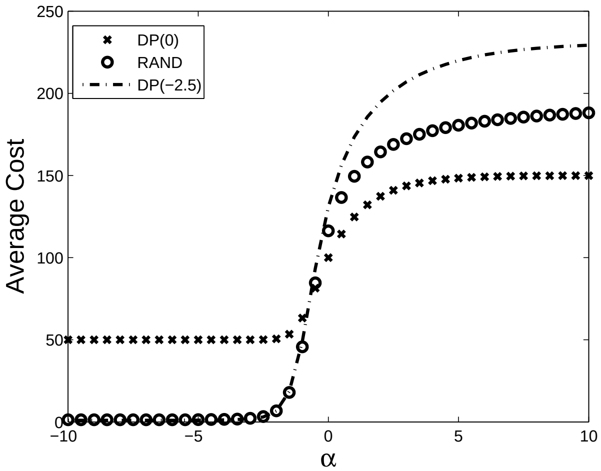

Figure 4. Performance of DP(0), DP(-2.5) and RAND as functions of α when $L = 100$.

location distribution. Let DP(α') denote the optimal (deterministic) average-cost strategy derived using dynamic programming when assuming $\alpha = \alpha'$ in the distribution of X. We then compute the expected search cost of DP(0) and DP(-2.5) when the location distribution is in fact defined by some other α, for $-10 \leq \alpha \leq 10$. Similarly, we calculate the average search cost under these distributions when using the optimal worst-case (randomized) strategy, RAND.

These results are shown in Figure 4, where the performance of these strategies are plotted for $L = 100$ as functions of α. As can be seen, DP(0) is more robust (less sensitive in the change in α) than RAND, while for DP(-2.5) the opposite is true. For small (negative) α, RAND outperforms DP(0) and in some cases the average-cost of DP(0) is 38 times larger. On the other hand, for large (positive) α, DP(0) is better, but the average-cost of RAND is greater only by a factor of 1.3. Thus we see that the dynamic programming strategy should only be used if we are fairly certain about the object location distribution.

This quantitative relationship obviously varies with the underlying assumptions on the location distribution and the errors introduced. This specific example nonetheless illustrates the general trade-off between search cost and robustness.

7 Conclusion

In this paper we study the class of TTL-based controlled flooding search methods used to locate an object/node in a large network. We derived an asymptotically optimal strategy whose search cost, under a general class of cost functions, is always within a factor of e of the cost of an omniscient observer. We examined its performance under other criterion, and discussed how the strategy's parameters can be adjusted to account for these new measures. These results are directly applicable in designing practical algorithms.

References

[1] Y. Baryshinikov, E. Coffman, P. Jelenkovic, P. Momcilovic, and D. Rubenstein. Flood search under the california split rule. *Operations Research Letters*, 32(3):199–206, May 2004.

[2] A. Borodin and R. El-Yaniv. *Online Comptutation and Competetive Analysis*. Cambridge University Press, 1998. Cambridge, UK.

[3] D. Braginsky and D.Estrin. Rumor routing algorithm for sensor networks. *Proc. International Conference on Distributed Computing Systems (ICDCS-22)*, 2002.

[4] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets. Manycast: Exploring the space between anycast and multicast in ad hoc networks. *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networks (MobiCOM'03)*, September 2003. San Diego, California.

[5] N. Chang and M. Liu. Controlled fboding search in a large network. *EECS Technical Report CGR 04-15*, 2004. University of Michigan, Ann Arbor.

[6] N. Chang and M. Liu. Revisiting the TTL-based controlled fboding search: Optimality and randomization. *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networks (MobiCOM'04)*, pages 85–99, September 2004. Philadelphia, PA.

[7] Z. Cheng and W. Heinzelman. Flooding strategy for target discovery in wireless networks. *Proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, Sept 2003.

[8] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, pages 153–181, 1999.

[9] S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, 1996. New York, USA.

[10] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM'99)*, August 1999. Seattle, Washington.

[11] S. Shakkottai. Asymptotics of query strategies over a sensor network. *Proceedings of IEEE Infocom*, March 2004. Hong Kong.

[12] J. Xie, R. Talpade, T. McAuley, and M. Liu. AMRoute: Ad Hoc Multicast Routing Protocol. *ACM Mobile Networks and Applications (MONET) Special Issue on Mobility of Systems, Users, Data and Computing*, 7(6):429–439, December 2002.