Online Searching*

Patrick Jaillet [‡] and Matthew Stafford[¶]

Abstract

We consider the problem of searching m branches which, with the exception of a common source s, are disjoint (hereafter called concurrent branches). A searcher, starting at s, must find a given "exit" t whose location, unknown to the searcher, is on one of the m branches. The problem is to find a strategy that minimizes the worst case ratio between the total distance traveled and the length of the shortest path from s to t. Additional information may be available to the searcher before he begins his search.

In addition to finding optimal or near optimal deterministic online algorithms for these problems this paper addresses the "value" of getting additional information before starting the search.

Keywords: online search, incomplete information, value of information

*Appeared in *Operations Research*, 49, 501--515, 2001. Research funded in part by ONR, grants N00014-94-1-0257 and N00014-96-1-0190, and NSF, grant DMI-9713682. Second author also wishes to thank SBC Technology Resources for its generous support.

[‡]MSIS Department and Civil Engineering Department, The University of Texas at Austin, Austin, TX 78712, USA, and Mathematics Department, ENPC, 75343 Paris cedex 07, France

[¶]SBC Technology Resources, Inc, 9505 Arboretum Blvd, Austin, TX 78759 USA, and Operations Research, The University of Texas at Austin, Austin, TX 78712, USA

1 Introduction

1.1 Motivation and overview

There are many situations in which present actions influence our "well being" in the future. The difficulty in these situations is that we have to make our decision based only on the past and the current task we have to perform. It is not even clear how to measure the quality of a proposed decision strategy. The approach usually taken is to devise some probabilistic model of the future and act on this basis. This is the starting point of the theory of Markov Decision Processes (see for example [11]).

The approach around which this paper is based is to compare the performance of a strategy that operates with *no knowledge of the future* (online) with the performance of the optimal *clairvoyant* strategy that has complete knowledge of the future (offline). This requires no probabilistic knowledge of the future and is therefore a "worst case" measure of quality. This new approach, first suggested in [18] and later called "competitive analysis" in [15], takes the following pessimistic approach in analyzing the performance of an online algorithm: an online algorithm is good only if its performance on *any* sequence of requests is within some (desired) factor of the performance of the offline algorithm. In particular, a good algorithm would certainly perform well in the presence of an unknown distribution.

In [6], the authors gave an abstract formulation (called task systems) and a formal definition for the study of the competitive analysis of online algorithms and problems. In [16] another abstract formulation, called k-server problems was introduced. More recently, in [7], the authors present the interesting notion of regret in the "online decision problem" setting.

Over the past ten years, online algorithms have received considerable research interest in computer science, and to a lesser extent in operations research. There are many interesting application areas which have been evaluated within this online framework. To include a few, let us mention paging, ressource allocation, scheduling, robotics, portfolio selection, and trading. In a recent book [5] one can find a nice introduction to the theory and applications of online computation and competitive analysis. For surveys aimed at the operations research and mathematical programming communities one can also consult [1, 2].

In this paper we consider generic searching problems in some given space. This situation does not necessarily involve sequential decisions over time but can nevertheless be included in the previous framework of online problems. Indeed, in many situations, the searcher has incomplete information about the space, and additional information can be acquired in a dynamic manner, as the search for a good path evolves.

Searching unknown graphs and planar regions is central to many areas of computer science and operations research. One classical application is in the area of navigation problems in robotics, where such problems come up repeatedly whenever a robot, exploring an unknown environment, faces an obstacle and tries to find the best way to avoid it.

Also many online graph searching problems come up persistently in the development of heuristics for intelligent searching and have come up repeatedly in the operations research literature (see [4, 8, 9, 10, 12]). In summary, these problems are simple and stylized versions of general problems for which an object or a boundary has to be located by successive moves in a largely unknown search space. Search problems generalize numerous other well-known online problems (e.g., metrical system and the k-server problem), and, for this reason, are key problems in the analysis of online strategies. Finally, "How much is it worth to have additional information before solving a problem?" is one of the major theoretical and practical motivations behind our line of research.

This paper will be concerned with the special case of searching m concurrent branches which, with the exception of a common source s, are disjoint. More precisely we will concentrate on the following problem: Given m concurrent branches (each a copy of R^+), a searcher, initially placed at the origin, has to find an "exit" which is at an unknown real distance $d \geq 1 > 0$ from the origin on one of the m concurrent branches. We will refer to this problem as the "m-concurrent branch" problem.

The article is organized around three main parts, the first two dealing with the case $m = 2$, and the third considering (the more challenging) generalizations to $m > 2$.

In the first section we assume that the searcher knows a priori that the exit is within a distance D from the source (hereafter called the bounded version as opposed to the general unbounded case). We develop and prove the optimality of a family of strategies that depends on D. A basic idea of the approach is to solve the following dual question: Given a competitive ratio r, what is the largest "extent" (i.e. the farthest we go in all directions) that can be searched without violating the ratio? We show that a strategy based on successive local optimization depending on r leads to the maximum extent, say $e(r)$. For any D, we then solve the primal searching problem by finding the smallest r such that $e(r) \geq D$. When D goes to infinity, this approach provides an optimal online strategy for the unbounded case with a competitive ratio of 9.

In the second part, we assume that the searcher receives "probabilistic information" of the following type: "The exit has a probability p_k of being on branch k". Based on dual arguments, we again propose and prove the optimality of deterministic online strategies for the 2-concurrent branch problem.

In the last part, we consider the extension of previous results to the m-concurrent branch problems, $m > 2$. These extensions are however not trivial and will require a more rigorous treatment of some arguments used in the 2-branch case. The reader can thus first see the main arguments in broad outline, maintaining greater intuition before finally taking on the most technical underpinnings.

1.2 Previous work and related problems

Baeza-Yates, Culberson, and Rawlins [3] discuss strategies for a problem very similar to our unbounded m-concurrent branch problem, in which the exit is at an integer distance from the origin. When $m = 2$, they give a proof that the strategy of alternatively moving on each branch, each time doubling the previous distance is optimal (among "monotone-increasing" strategies) with a competitive ratio of 9. When $m > 2$, they propose the following strategy: move in the integrally increasing powers of $m/(m-1)$ in a cyclic manner, visiting branches in the same order over and over again. They argue that, among all "monotone-increasing" cyclic strategies, this one is optimal. In Kao, Reif and Tate [14], the authors present an optimal randomized algorithm for what can be defined to be our 2-concurrent branch problem.

A different but related problem has also appeared in the literature under the name of the "Layered Graph Traversal" problem. A *layered graph* is a connected weighted graph whose nodes are partitioned into sets (i.e., layers) $L_0 = \{s\}, L_1, L_2, \ldots$ and all edges connect nodes in consecutive layers. The edges between layer L_i and layer L_{i+1} are *all revealed* when the searcher visits some nodes in L_i (this is the main difference with our problems). This problem is introduced in Papadimitriou and Yannakakis [17] and is solved optimally for the case of 2 disjoint paths, by using the results of [3].

The fundamental contributions of our paper are:

1. A rigorous proof that the strategies introduced in [3] are optimal amongst all possible strategies. Baeza-Yates et al. only consider the restricted class of "monotone-increasing" cyclic strategies. Proving that there is no loss of optimality in restricting to this class is not difficult in the 2-branch case but becomes highly non-trivial for $m > 2$.

2. A framework, using duality and mathematical programming concepts, allowing the following rigorous extensions of the "prototype" result given by Baeza-Yates et al. for $m = 2$:

   (a) For $m > 2$, the optimal competitive ratio for the infinite-extent m-branch problem is $R_m := 2m[m/(m-1)]^{m-1} + 1$.

   (b) (Value of additional deterministic information about the target location.) For $m = 2$, we quantify the rate at which $e(r)$, the maximum searchable extent subject to competitive ratio r, approaches infinity as r approaches 9 from below. For $m > 2$, we give a lower bound for the same rate as $r \uparrow R_m$ (the value of R_m is as given in the previous paragraph.)

   (c) (Value of probabilistic information about the target location.) For $m = 2$, suppose we are given p, the probability that the target location is somewhere on branch 1. We analyze the optimal competitive ratio as a function of p.

To be fair, it is possible to extend the $m = 2$ result of Baeza-Yates et al. directly (i.e. without appealing to our mathematical programming/duality framework) to the case where the distance from source to target is real (recall that [3] assumes this distance is an integer.) This can be accomplished in three steps:

1. Extend their result to the case where said distance has (integer) *lower* bound $d > 1$.

2. Note the essential equivalence of the original problem with allowable source-to-target distances on the lattice $\{j/2^k\}_{j \geq 1}$. (The integer $k \geq 1$ is fixed but arbitrary.)

3. Combine the two previous items via a scaling argument yielding the same competitive ratio of 9 whenever the source-to-target distance is on the lattice $\{j/2^k\}_{j \geq 2^k}$. Then one can fashion a limiting argument as k approaches infinity.

None of the steps in the above outline is particularly difficult. However, when combined with the argument of Baeza-Yates et al., the resulting proof is laborious. Further, we do not see how to duplicate the main results of this paper using this sort of argument.

1.3 Notation and terminology

Before presenting our results, one needs to understand how to evaluate the quality of strategies for online problems. Let $\mathcal{S}$ be a deterministic strategy for the m-concurrent branch problem. For any position of the exit (specified by the pair (b, d), indicating that the exit is at a distance $d \geq 1$ from the origin on branch b), let $cost_{\mathcal{S}}(b, d)$ be the cost incurred (under strategy $\mathcal{S}$) to find it. Throughout this paper, our cost function is total distance traveled. Other choices include elapsed time, which becomes interesting for parallel searching (see [13]).

The *competitive ratio* of the deterministic strategy $\mathcal{S}$ is defined to be

$$\sup_{(b,d)} \{cost_{\mathcal{S}}(b, d)/d\}$$

A strategy (and its competitive ratio) is said to be *optimal* if it has the smallest provable competitive ratio.

2 The 2-Concurrent Branch Problem With Deterministic Information

We assume here that we have two branches, numbered 1 and 2, meeting at the origin; each is a copy of $R^+ = [0, \infty)$. Any deterministic search strategy of these two branches can then be defined as an infinite sequence of real numbers $\{x_i\}_{i \geq 1}$, where x_i is the distance between the origin and the returning point during the i^{th} "attempt" (an attempt being defined as leaving the origin on one branch, exploring this branch up to a returning point - possibly infinite, and, in case of a finite returning point, returning to the origin). It is clear that any sensible strategy will alternate on the two branches, so that if the 1^{st} attempt is on Branch 1, then any i^{th} attempt with an odd i will also be on Branch 1. Let the "extent" of a strategy be defined as the set of points on the two branches within a distance $\min\{\max_{i\ odd} x_i; \max_{i\ even} x_i\}$ from the origin (note that the extent of a strategy would be the set of all points on the two branches if the sequence $\{x_i\}_i$ goes to infinity for both odd and even indices). At times it will be convenient to blur the distinction between the extent of a strategy as defined here and the quantity $\min\{\max_{i\ odd} x_i; \max_{i\ even} x_i\}$ itself; the distinction should be clear from context.

2.1 Unbounded case

Here let us assume that the position of the exit is not a priori bounded. The specific strategy analyzed in [3] (see 1.2 above) can be described by the sequence $x_i = 2^i$. The extent of such a strategy is the entire set of points on the two branches. It is also easy to calculate its competitive ratio. If the exit is say at a distance d on Branch 1, then this strategy will discover it on the attempt $2k+1$, where $2^{2k-1} < d$ and $2^{2k+1} \geq d$. The total distance covered will then be $2\sum_{j=1}^{2k} 2^j + d = 4(2^{2k}-1) + d$. The ratio of this total distance to the distance d would be in the worst limiting case $4(2^{2k}-1)/(2^{2k-1}) + 1$ [the worst case corresponds to the unexplored point on Branch 1 closest to the origin, i.e. when $d \downarrow 2^{2k-1}$]. We then have

$$\sup_k 4(2^{2k}-1)/2^{2k-1} + 1 = 4 * 2 + 1 = 9.$$

Calculations are very similar if the exit is on Branch 2. The overall competitive ratio of the strategy $(2^i)_{i\geq 1}$ is thus 9. We will show below that this is an optimal competitive ratio, in the sense that no other strategy will provide a lower competitive ratio.

2.2 Bounded case

If it is known a priori that the exit is *exactly* at a distance $D \geq 1$ from the origin, the optimal competitive ratio is easily seen to be 3. Indeed the optimal strategy is to go on Branch 1 up to the distance D and then go back to the origin and go on Branch 2 up to D. In the worst case the exit would be on Branch 2 and the competitive ratio would be $(2D+D)/D$.

Let us now consider the case where the exit is known a priori to be *within* a distance $D \geq 1$ from the origin. Consider the following *dual* problem: Given a ratio $r > 3$, find a strategy that maximizes the extent searched without violating this ratio (i.e a strategy $\mathcal{S}$ such that $cost_{\mathcal{S}}(b,d) \leq rd$ for all points (b,d) in the extent of $\mathcal{S}$).

For this problem, suppose that after $n-1$ attempts, the searcher is at a distance x_{n-1} from the origin on one (say, Branch 1) of the two branches, and then turns back. As the searcher decides on the distance x_n on Branch 2, the *critical* point (i.e. with maximum ratio) limiting the value of x_n will be at a distance $x_{n-1}+\varepsilon$ on Branch 1, for any ε arbitrarily small [i.e., will be the unexplored point of Branch 1 closest to the origin]. We then must have $(2x_1 + \cdots + 2x_n + x_{n-1})/x_{n-1} \leq r$, or

$$x_1 + \cdots + x_n \leq \frac{r-1}{2} x_{n-1}. \qquad (I_n)$$

We will refer to this inequality with n as a variable index, i.e. I_k means the same inequality with n replaced by k, and so on. Suppose that for all $n \geq 1$, the searcher chooses to go to the maximum possible value of x_n, as defined above. This policy of maximizing x_n at each attempt leads to a strategy with maximum extent. In order to see that, consider the successive solution of the following linear programming problems (P_k), for increasing k, starting with $k = 1$. For convenience, we introduce the quantity $\rho := (r-1)/2$.

$$\begin{array}{ll} \text{Maximize} & x_k \\ \text{subject to} & x_1 \leq \rho = (r-1)/2 \\ & x_1 + x_2 \leq \rho x_1 \\ & \dots \\ & x_1 + \cdots + x_k \leq \rho x_{k-1} \\ & 1 \leq x_1 \leq \cdots \leq x_k \end{array} \qquad (P_k)$$

Let x_k^* be the optimal value of (P_k). We will show that the solution to our previous dual problem [i.e., the problem of maximizing the extent given a ratio $r > 3$] corresponds to x_q^*, where q is the largest integer (possibly infinite) such that $x_1^* \leq x_2^* \leq \cdots \leq x_q^*$.

In terms of the Problem (P_k), the strategy of maximizing x_n at each attempt corresponds to making all upper bound constraints tight. Here is how we argue that successive maximization is optimal. For given values of $\{x_1, \dots x_{k-1}\}$, x_k is maximized when (I_k) is "binding". In that case we have

$$\begin{aligned} x_k &= \rho x_{k-1} - (x_1 + \cdots + x_{k-1}) \\ &= (\rho - 1)x_{k-1} - (x_1 + \cdots + x_{k-2}) \end{aligned} \qquad (1)$$

providing a functional relationship between x_k and $\{x_1, \dots x_{k-1}\}$. This is true for *any* choice of $\{x_1, \dots x_{k-1}\}$. Now since $r > 3$, $\rho - 1 = (r-3)/2$ is positive. So x_k is in fact *increasing* when viewed as a function of x_{k-1} alone. Therefore the Problem (P_k) with $x_1 + \cdots + x_k = \rho x_{k-1}$ reduces to maximizing x_{k-1} with respect to the other constraints, and this problem is (P_{k-1}), and a classical induction argument follows.

Now that we have settled on a policy of successive maximization, (1) describes the general term in a recursion that begins with $x_1 = \rho$. From the first form of (1) one obtains a simplified linear recursive relationship for $k \geq 3$:

$$x_k = \rho(x_{k-1} - x_{k-2}). \qquad (2)$$

If we reach a value of k for which $\rho(x_{k-1} - x_{k-2})$ fails to exceed x_{k-1}, then using (2) to assign x_k and subsequently x_{k+1} will result in a non-positive value for x_{k+1} [indicating, in light of our successive maximization optimality argument, that no further progress can be made beyond the k^{th} attempt while adhering to the target ratio.] This is true all the more if x_{k-1} fails to exceed x_{k-2} [a case that only has to be considered separately for $k = 3$ and ρ sufficiently small.]

Therefore we should stop searching as soon as the linear relationship (2) leads to $x_{k-1} \leq x_{k-2}$ – in that case one should define x_j to be 0 for all $j \geq k$. The following theorem ties up all these cases in a neat package.

Theorem 1. *Let $\rho = (r-1)/2$. Given a competitive ratio $r \geq 3$, the maximum possible extent corresponds to the following optimal strategy: $x_1 = \rho$, $x_2 = \max\{x_1; x_1(r-3)/2\}$, and for $n \geq 3$,*

$$x_n = \begin{cases} \max\{x_{n-1}; \rho(x_{n-1} - x_{n-2})\} & \text{if } x_{n-1} > x_{n-2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Crucial to the analysis of the recursive relationship (3) is the associated characteristic equation $\xi^2 = \rho(\xi - 1)$. The discriminant $\Delta = \rho^2 - 4\rho$ of this equation needs to be nonnegative for the corresponding strategy to have an extent covering the entire set of points of the two branches. In that case the solutions of the characteristic equation are given by

$$\xi = \frac{\rho \pm \sqrt{\rho^2 - 4\rho}}{2}. \tag{4}$$

Note that $\Delta \geq 0$ implies that $\rho \geq 4$ and thus that $\xi_1 \stackrel{\text{def}}{=} (\rho + \sqrt{\rho^2 - 4\rho})/2 \geq 2 > 1$. Hence $\Delta \geq 0$ is a necessary and sufficient condition for the corresponding strategy to have an extent covering the entire set of points of the two branches.

When $r = 9$ or $\rho = 4$ (which is the smallest feasible value for r in order to cover the entire two branches), the unique root is $\xi_1 = \rho/2 = 2$, and we get $x_n = (a + bn)\xi_1^n = (a + bn)2^n$ for the generic solution. From $x_1 = \rho = 4$ and $x_2 = \rho(\rho - 1) = 12$, we conclude that $a = 1$ and $b = 1$, and that $x_n = (1 + n)2^n$ which is, up to a factor $(1 + n)$, a result similar to the one obtained by [3]. Note that the two strategies $\{2^n\}_n$ and $\{(1 + n)2^n\}_n$ are equivalent (same competitive ratio), but that the latter will cover more space at each successive attempt. We can summarize our results with the following corollary, which can put the unbounded case to rest.

Corollary 1. *The maximum possible extent includes the entire branches if and only if $r \geq 9$. For $r = 9$ the previous optimal strategy corresponds to $x_n = (n + 1)2^n$, $n \geq 1$.*

When $r < 9$ the calculations are more tedious, but remain trivial. Now Equation (4) describes a complex pair of modulus $\rho^{1/2}$; therefore we can compactly write the characteristic roots in the form $\rho^{1/2}e^{\pm i\psi}$. For convenience, we let $\beta = \big((4-\rho)/\rho\big)^{1/2} = \big((9-r)/(r-1)\big)^{1/2}$; then it is easy to see that the argument ψ above is given by $\arctan\beta$. It is also easy to check (we omit the details) that the unique solution to the linear recursion (2) satisfying the initial conditions $x_1 = \rho$ and $x_2 = \rho(\rho-1)$ is

$$\frac{1}{2}\big(1 - \frac{1}{\beta}i\big)\rho^{k/2}e^{ik\psi} + \frac{1}{2}\big(1 + \frac{1}{\beta}i\big)\rho^{k/2}e^{-ik\psi},\ k \geq 1.$$

From Theorem 1 one needs to find q, the largest k for which $x_1 \leq x_2 \leq \cdots \leq x_k$. Looking ahead to Expression (5), in which we have simplified the above formula and converted to polar coordinates, it is clear that $q = \lfloor \pi/\psi \rfloor - 2$. In the statement of the next corollary, we rename this quantity $n^*(r)$ to emphasize the dependence of the "stopping index" q on the ratio r.

Corollary 2. *For $3 \leq r < 9$, let $\rho = (r-1)/2$, $\beta = ((9-r)/(r-1))^{1/2}$, $\psi = \arctan\beta$, and $n^*(r) = \lfloor \pi/\psi \rfloor - 2$. The optimal strategy is*

$$x_n = \rho^{n/2}\left(\cos n\psi + \frac{\sin n\psi}{\beta}\right) \text{ for } 1 \leq n \leq n^*(r), \tag{5}$$

$x_{n^*(r)+1} = x_{n^*(r)}$, and $x_n = 0$ otherwise. The bound defining the maximum extent is defined by $\mathrm{e}(r) = x_{n^*(r)}$.

Finally one can conclude with the following overall consequence:

Theorem 2. *Given we know that the exit is within a distance $D \geq 1$ from the origin, the optimal competitive ratio will be $r^* = \inf\{r : \mathrm{e}(r) \geq D\}$. The corresponding optimal search strategy is defined in the previous corollary with $r = r^*$.*

In the following table, we compare the extent of the "best power 2" strategy (i.e., of the form $x_i = \rho 2^{i-1}$, where $\rho = (r-1)/2$) with the extent of the previous optimal strategy, for some values of the competitive ratio $r < 9$.

	Extent	
Ratio	Power 2	Optimal
3.00	1	1
5.00	2	2
7.00	6	9
8.00	14	69.67
8.50	30	1,050.81
8.90	126.4	5.9×10^7
8.99	1022.72	1.3×10^{26}

Note that when $r \to 9$, one can calculate the rate of convergence of the extents to infinity.

For the the "best power 2" strategy, the extent, when $3 \le r < 9$, is given by $\rho 2^{k-1}$, where k is the smallest of the integer $i \ge 1$ such that

$$\frac{2(1+2+\cdots+2^i)+2^{i-1}}{2^{i-1}} = 9 - \frac{4}{2^i} > r.$$

This implies that $k = \max\left\{1, \lfloor \log \frac{4}{9-r} \rfloor + 1\right\}$. One can easily deduce that for $5 \le r < 9$, the extent is bounded as follows

$$\frac{r-1}{9-r} \le e_{power}(r) < 2\frac{r-1}{9-r},$$

which, in turn, implies that $e_{power}(r) = \Theta(\frac{1}{9-r})$ when $r \to 9$.

For the optimal strategy, based on Corollary 2, $e_{opt}(r) = x_{n^*(r)}$, where $n^*(r) = \lfloor \pi/\psi \rfloor - 2$, and $\psi = \arctan\left[(\frac{9-r}{r-1})^{1/2}\right]$. Also we have

$$1 \le \cos n^*(r)\psi + \frac{\sin n^*(r)\psi}{\beta} \le 2,$$

so that we have $\rho^{n^*(r)/2} \le e_{opt}(r) \le 2\rho^{n^*(r)/2}$. We can then conclude that $e_{opt}(r) = \Theta(2^{c/(9-r)^{1/2}})$.

In the following table we have listed the optimal competitive ratio $r^*(D)$ for some values of D.

Distance D	Optimal $r^*(D)$
1	3.00
5	6.38
10	7.06
100	8.10
1000	8.49
10000	8.68
∞	9.00

3 The 2-Concurrent Branch Problem With Probabilistic Information

Let us return to the case of the unbounded 2-concurrent branch problem and assume now that the searcher is told that the probability of the exit being on Branch 1 is $p \geq 1/2$. How to use this information? First of all, if $p > 1/2$ then the problem is not symmetric and it is quite natural to see that an optimal strategy will have to start on Branch 1 (as opposed to the previous sections for which the total symmetry implied indifference about which branch to search first). In order to solve this problem, we are going to extend the dual idea proposed in the previous section. Namely, let r_1 and r_2 be two given ratios not to violate on Branch 1 and Branch 2, respectively. We are interested in such pair of ratios for which there exists at least a feasible search strategy whose extent is the set of all points of the two branches. In other words, for such a strategy, given that the exit is on Branch 1, the conditional competitive ratio would be r_1, and the conditional competitive ratio given the exit is on Branch 2 would be r_2. The overall expected competitive ratio will then be $pr_1 + (1-p)r_2$.

If $p = 1/2$, then we have $r_1 = r_2 = r$, and the best we can do (smallest r we can choose in order to get an infinite extent) is $r = 9$ (see previous section). If $p > 1/2$, r_1 and r_2 can't be both smaller than 9 (for a strategy to have an infinite extent). It is also clear that we will want to visit more of Branch 1 than Branch 2, and thus that we should allow a larger ratio on Branch 2.

Given r_1 and r_2, a feasible strategy $\{x_i\}_i$ will need to verify:

$$
\begin{aligned}
&2x_1 + 1 \leq r_2 \\
&2x_1 + 2x_2 + x_1 \leq r_1 x_1 \\
&\dots \\
&2x_1 + \dots + 2x_{2k+1} + x_{2k} \leq r_2 x_{2k} \\
&2x_1 + \dots + 2x_{2k+2} + x_{2k+1} \leq r_1 x_{2k+1} \\
&\dots
\end{aligned}
$$

Using the same arguments as in Section 2 the local strategy of maximizing the distance covered at each attempt can be shown to be globally optimal (i.e. leading to the largest extent). This is equivalent to turning the above inequalities into equalities.

Let $\rho_1 = (r_1 - 1)/2$ and $\rho_2 = (r_2 - 1)/2$. We then get:

$$
\begin{aligned}
&x_1 = \rho_2 \\
&x_2 = \rho_1 x_1 - \rho_2 \\
&\ldots \\
&x_{2k+1} = \rho_2 x_{2k} - \rho_1 x_{2k-1} \\
&x_{2k+2} = \rho_1 x_{2k+1} - \rho_2 x_{2k} \\
&\ldots
\end{aligned}
$$

By defining $x_0 = 1$ and $x_{-1} = 0$, we then have for all $i \geq 1$,

$$
\begin{aligned}
x_{2i-1} &= \rho_2 x_{2i-2} - \rho_1 x_{2i-3} \\
x_{2i} &= \rho_1 x_{2i-1} - \rho_2 x_{2i-2}.
\end{aligned}
$$

For a given search strategy $\{x_i\}_i$, let $\{y_i\}_i$ be the subsequence for Branch 1 and $\{z_i\}_i$ for Branch 2 ($y_i = x_{2i-1}$ and $z_i = x_{2i}$). The previous equations can be rewritten as follows:

For all $i \geq 1$,

$$
\begin{aligned}
y_i &= \rho_2 z_{i-1} - \rho_1 y_{i-1} \\
z_i &= \rho_1 y_i - \rho_2 z_{i-1}.
\end{aligned}
$$

It implies that both y and z follow the same recursive relationships:

$$
\begin{aligned}
y_i &= [\rho_1\rho_2 - (\rho_1 + \rho_2)] y_{i-1} - \rho_1\rho_2 y_{i-2} \\
z_i &= [\rho_1\rho_2 - (\rho_1 + \rho_2)] z_{i-1} - \rho_1\rho_2 z_{i-2}.
\end{aligned}
$$

The corresponding characteristic equation is $\xi^2 = [\rho_1\rho_2 - (\rho_1 + \rho_2)]\xi - \rho_1\rho_2$. In order for the strategy $\{x_i\}_i$ to have an infinite extent the discriminant of this characteristic equation needs to be nonnegative. Let $f(\rho_1, \rho_2) = (\rho_1\rho_2 - \rho_1 - \rho_2)^2 - 4\rho_1\rho_2$ be this discriminant. When $f(\rho_1, \rho_2) \geq 0$ the largest solution (the unique solution if the discriminant is zero) of the characteristic equation is given by

$$
g(\rho_1, \rho_2) = \frac{(\rho_1\rho_2 - \rho_1 - \rho_2) + \sqrt{(\rho_1\rho_2 - \rho_1 - \rho_2)^2 - 4\rho_1\rho_2}}{2}.
$$

In order to have an infinite extent we need to have $g(\rho_1, \rho_2) > 1$.

The optimal competitive ratio under such probabilistic information is thus given by the optimal value of the following mathematical programming problem:

$$
\begin{aligned}
\text{Minimize} \quad & p\rho_1 + (1-p)\rho_2 \\
\text{subject to} \quad & f(\rho_1, \rho_2) \geq 0 \\
& g(\rho_1, \rho_2) > 1 \\
& 1 \leq \rho_1 \leq 4 \\
& \rho_2 \geq 4
\end{aligned}
$$

Ignoring first the constraint $g(\rho_1, \rho_2) > 1$, it is not hard to see that the optimization will always happen on the boundary $f(\rho_1, \rho_2) = 0$. Now $f(\rho_1, \rho_2) = 0$, $1 \leq \rho_1 \leq 4$, $\rho_2 \geq 4$ together imply that $g(\rho_1, \rho_2) > 1$. The mathematical programming problem can thus be simplified to:

$$\begin{array}{ll} \text{Minimize} & p\rho_1 + (1-p)\rho_2 \\ \text{subject to} & f(\rho_1, \rho_2) = 0 \\ & 1 \leq \rho_1 \leq 4 \\ & \rho_2 \geq 4 \end{array}$$

The corresponding optimal deterministic strategy consists of starting on Branch 1, and subsequently maximizing the distance at each attempt subject to not exceeding the competitive ratios r_1^* on Branch 1 and r_2^* on Branch 2, where r_1^* and r_2^* are the solutions of the previous mathematical programming.

In the following table we have listed the optimal competitive ratios for some value of p.

Probability p	r_1^*	r_2^*	Optimal ratio
0.99	3.96	64.34	4.56
0.90	5.39	19.97	6.84
0.80	6.31	14.39	7.93
0.70	7.15	11.82	8.55
0.60	8.02	10.20	8.89
0.50	9.00	9.00	9.00

4 Extension to $m > 2$ Branches

We return to the case where the searcher has no *a priori* information and extend our results to encompass 3 or more branches. By "no information" we mean that the searcher has no probability distribution saying that one branch is more likely than another; nor is an upper bound on the distance of the exit from the source given.

In the 2-branch case, it is easy to see that the searcher should alternate between the branches. The obvious generalization of this to the case $m > 2$ is the concept of a cyclic search (here, the branches are repeatedly visited in the same order - no branch is visited twice between consecutive visits to any other branch.) Moreover, the symmetry of the problem and the minimax criterion for evaluating a strategy make the idea of a cyclic search appealing.

As it turns out, one has to work surprisingly hard to show that the class of cyclic searches contains an optimal strategy. We choose to do this by first showing that one can assume, without loss of generality, that the sequence of search extents is non-decreasing.

Suppose for a moment that this property of non-decreasing searches has been established and that the searcher has returned to the source, ready to begin the next excursion. We then argue that the searcher should take this excursion on the branch whose explored extent, up to the current juncture, is smallest. This "least-searched-so-far" discipline is the link between the non-decreasing extent sequence and the cyclic property we seek.

Once we know that there is a cyclic strategy which is optimal, we adapt the solution technique of earlier sections. One can identify a recurrence similar to that of Section 2.2 easily enough. The characteristic polynomial of this recurrence yields a putative minimum value of the competitive ratio for which the infinite-extent problem is solvable. *Proving* that, below this candidate ratio, the infinite-extent problem is infeasible turns out to be subtle and involved here. This is in marked contrast to the two-branch case, where this infeasibility comes quite naturally via the induction argument presented in Section 2.2.

4.1 Fundamentals and Notation

We take this opportunity to introduce a change of notation, or more accurately a change of subscripting. Statements like "odd-numbered searches explore branch 1" are most easily generalized using modular arithmetic: "search i explores branch $i \pmod m$". This in turn means that our numbering schemes should start at 0: the search extent sequence begins with x_0, and the branches are labeled $0, 1, \ldots, m-1$.

Definition 1. A (deterministic) *strategy* is, formally speaking, a map from the non-negative reals R^+ into the search space X. For a strategy $\mathcal{S} : R^+ \to X$, $\mathcal{S}(d)$ is interpreted as "the position of the searcher in the space X after d total units of distance have been traversed."

The *discovery map* $\mathrm{disc}_{\mathcal{S}} = \mathrm{disc} : X \to R^+$ is a sort of inverse to $\mathcal{S}$. For $x \in X, \mathrm{disc}(x) :=$ the smallest distance d for which $\mathcal{S}(d) = x$. If strategy $\mathcal{S}$ never visits $x, \mathrm{disc}(x)$ is taken to be infinite.

The discovery map is synonymous with the cost function of earlier sections. We use this different terminology to emphasize the view of a strategy as a function and the discovery map as its "one-sided inverse."

Definition 2. The *competitive ratio map* $r = r_{\mathcal{S}} : X \to R^+ \cup \{\infty\}$ is given by $r(x) = \frac{\mathrm{disc}(x)}{\|x\|}$. The denominator $\|x\|$ represents the distance from the source to x. This map compares the distance traveled under strategy $\mathcal{S}$

(to reach an exit placed at x) to the distance that would be traveled if the searcher knew which branch to explore.

The competitive ratio $r_{\mathcal{S}}$ of a strategy $\mathcal{S}$ is $\sup_{\{x\in X \mid \|x\|\geq 1\}} \frac{\mathrm{disc}(x)}{\|x\|}$; this is a "worst-case" measure of the performance of $\mathcal{S}$.

Remark 1. If the distance of the exit from the source is not bounded away from 0, no strategy can have finite competitive ratio. The imposition of a bound thus allows us to use this ratio as a metric for meaningful comparisons between strategies. The fact that we take this lower bound to be 1 (rather than ε, say) is largely a matter of convenience. Converting to a different bound amounts to rescaling our measure of distance (and if one strategy has better competitive ratio than another *before* a uniform rescaling, then the same is true *after* the rescaling.)

The space of *all* search strategies is huge. We want to confine our attention to a tractable subspace (without discarding all optimal-ratio strategies in so doing.) This reduction to a "reasonable" strategy space takes place in several stages, the first of which is embodied in the following observation.

Observation. *The searcher should never retreat towards the source unless it is in the process of returning to the source to search another branch. Indeed, a "partial return" increases the discovery map of every heretofore undiscovered point, without decreasing the discovery map of any point. Further, the searcher must (always) eventually return to the source in any strategy whose competitive ratio is finite.*

Suppose we graph the distance of the searcher from the origin as a function of total distance traveled. This observation says that the resulting graph should be a sequence of isosceles triangles, each of which has base along the d-axis. The ascending sides of these triangles will each have slope +1, representing travel away from the origin. The descending sides of these triangles will each have slope -1. Consecutive triangles should abut, i.e. the graph has no horizontal segments at all.

Definition 3. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of altitudes of the isosceles triangles just described. This will be called the sequence of *search extents*; the indexing will be derived in the order in which the searches depart the origin. We will have occasion to use the abbreviation $\{x_k\}$ for the (entire) search extent sequence; when referring to a subsequence, we will always specify the range of indices explicitly.

Standing Assumptions. In what follows, we consider only searches where all extents are finite. This inflicts no loss of generality in the sense that it discards no strategies whose competitive ratios are finite. We also restrict ourselves to strategies where all extents are greater than or equal to 1. (Deletion of any extent smaller than 1 from the sequence does not increase the discovery map of any point and thus cannot increase the associated competitive ratio.)

Definition 4. A strategy $\mathcal{S}$ is *ergodic* if it explores the entire search space. In view of our standing assumptions, suppose we look at the subsequence of the extent sequence $\{x_k\}$ corresponding to all of the searches of a given branch. Ergodicity says that this subsequence is unbounded, regardless of which branch we have selected. This in turn implies that each branch is visited infinitely often.

Definition 5. *Dominating Strategy.* Given two strategies $\mathcal{S}$ and $\mathcal{T}$, $\mathcal{T}$ is said to *dominate* $\mathcal{S}$ if $r_{\mathcal{T}} \leq r_{\mathcal{S}}$.

Claim. *Every strategy is dominated by an ergodic strategy.*

No non-ergodic strategy can have finite competitive ratio, since its discovery map must be infinite somewhere. We have exhibited a finite-ratio strategy for the 2-branch case, and will do so later for the general m-branch case. So the claim is verified.

4.2 Restricting to the Non-Decreasing Space of Strategies

The competitive ratio map is an unwieldy object. We will eliminate some pathologies by observing that we can further restrict the space of strategies we have to consider. Once this is done, we will be able to identify a sequence of local suprema for the competitive ratio map. This sequence will be an important tool in what follows.

4.2.1 The Branch and Next Sequences; Progressive Strategies

We can describe a search strategy by specifying two sequences. The first of these is the sequence of search extents introduced earlier. If for all k we also indicate $b(k)$, the branch explored by the k^{th} search, a strategy is completely determined. We find it more convenient to denote the branch sequence using functional notation (rather than subscripts). For each $k \geq 0$, let $n(k)$ be the index of the next search to visit branch $b(k)$. To accustom the reader to the notation, we observe that $b(n(k)) = b(k)$ for all k. If we permute the numbering of the branches, $n(\cdot)$ does not change, nor does the competitive ratio of the corresponding strategy. Up to such a permutation, the sequences $\{x_k\}$ and $n(\cdot)$ completely determine a strategy.

Definition 6. A search strategy $\mathcal{S}$ is *progressive* provided that it is

1. *branch-increasing*, i.e. $x_j < x_{n(j)}$ for all j.
2. *non-repetitive*, i.e. $n(j) > j + 1$ for all j.
3. ergodic. This implies in particular that $n(j) < \infty$ for all j.

Claim. *Every strategy is dominated by a progressive strategy.*

The claim is easy to verify. If $x_{n(k)} < x_k$ we can remove $x_{n(k)}$ from the extent sequence (while leaving the branch assignments of all remaining searches unchanged); this does not increase the discovery map at any point.

Similarly, if $b(k) = b(k+1)$ then the smaller of x_k, x_{k+1} can be deleted from the extent sequence (again without degrading our performance measure.) These two types of "pruning" should be performed in the order introduced: suppose we start with an ergodic strategy and convert it to a branch-increasing search, pruning as indicated in the previous paragraph. The repetition-eliminating pruning of the current paragraph will not destroy the branch-increasing property thus established. This proves the claim.

Definition 7. *(Sequence of Competitive Ratios)* For a progressive strategy with extent sequence $\{x_k\}$, the k^{th} competitive ratio, is given by

$$r_k := 1 + \frac{2(x_0 + x_1 + \ldots + x_{n(k)-1})}{x_k}$$

Having defined r_k for $k \geq 0$, we also give a value for r_{-1}. Let $\alpha(b)$ denote the (index of) the first search to explore branch b. Then

$$r_{-1} := \max_b (1 + 2(x_0 + x_1 + \ldots + x_{\alpha(b)-1})).$$

The sense of this definition is revealed by noting that, for small ε, the $n(k)^{th}$ search is the one that discovers an exit located on branch $b(k)$ at distance $x_k+\varepsilon$ from the source. (It is the branch-increasing property that allows us to make this claim.) Let t_ε denote this exit location. The numerator in the expression given is the total distance traveled up to the commencement of the $n(k)^{th}$ search. Adding $x_k + \varepsilon$ to this numerator and dividing by $x_k + \varepsilon$ therefore calculates $r(t_\varepsilon)$, the ratio map evaluated at t_ε. Clearly this map increases as $\varepsilon \searrow 0$; thus the limit, which we are calling r_k, is a local least upper bound for the competitive ratio map. Thus r_k is, intuitively, the competitive ratio for an exit location "just past" the turning-back point of the k^{th} search. Using the compact notation x_k^+ for distances "just past" the k^{th}

turning-back point, we can summarize that r_k is really a limit by saying that r_k is associated with exit location x_k^+ on branch $b(k)$.

The presence of r_{-1} in the extent ratio sequence accounts for exit locations at extent 1^+ on any given branch.

Note that the inverse mapping $n^{-1}(j)$ is not defined whenever search j is the first excursion to branch $b(j)$. In this case, we perpetrate a slight abuse of notation: let $x_{n^{-1}(j)} := 1$, and think of $r_{n^{-1}(j)}$ as the competitive ratio for an exit location at distance 1^+ from the origin on branch $b(j)$. We will need this notational convenience in the proof of Theorem 3.

The exposition following Definition 7 makes it clear that we can assess the performance of a progressive strategy by looking at its competitive ratio sequence (in preference to working directly with the ratio map.) We now formalize this for future reference.

Observation. *For a progressive strategy $\mathcal{S}$,*

$$r_{\mathcal{S}} = sup_{k \geq -1}\{r_k\}.$$

Theorem 3. *We can assume without loss of generality that the sequence $\{x_k\}$ of search extents is non-decreasing: any strategy $\mathcal{S}$ for which this is not the case is dominated by some progressive strategy $\mathcal{S}'$ for which the extent sequence is non-decreasing.*

The proof is somewhat long, with numerous details that have to be handled carefully. However, the main idea is quite simple: where a large search extent precedes a small one, perform an interchange. We must then amend the sequence $n(\cdot)$ of "next pointers" in order to keep competitive ratios under control. The matter is somewhat delicate, but can be resolved by taking care to retain the branch-increasing property. We provide these details in the following lemma.

Lemma 1. *Suppose $\mathcal{S}$ is a progressive strategy for which $x_j > x_{j+1}$. Then there exists a dominating strategy $\mathcal{S}'$ with extent sequence*

$$x_0, x_1, \ldots x_{j-2}, x_{j-1}, x_{j+1}, x_j, x_{j+2}, x_{j+3}, \ldots$$

Proof. We use primes throughout to distinguish features of the new strategy $\mathcal{S}'$ from those of $\mathcal{S}$. We set $x'_j = x_{j+1}$ and $x'_{j+1} = x_j$; for all $k \neq j$ or $j+1$, let x'_k and x_k be identical.

Two cases must be considered.

Case I: $x_{j+1} > x_{n^{-1}(j)}$. Construct $\mathcal{S}'$ by setting $n'(j) = n(j+1)$ and $n'(j+1) = n(j)$; for all other indices, the $n(\cdot)$ pointers are unchanged.

We have

$$r_j = 1 + \frac{2(x_1 + \ldots + x_{n(j)-1})}{x_j} \quad \text{and} \tag{6}$$

$$r'_{j+1} = 1 + \frac{2(x_1 + \ldots + x_{n'(j+1)-1})}{x'_{j+1}}. \tag{7}$$

These two quantities are the same by construction. The underlying observation is that both numerators must contain x_j and x_{j+1}, since for a progressive strategy $n(j) > j + 1$. Similarly, $r_{j+1} = r'_j$. To verify this fact, one only need remove the primes from (7) and insert primes in corresponding places in (6).

The ratios for searches $n^{-1}(j)$ and $n^{-1}(j + 1)$ must also be checked:

$$\begin{aligned} r_{n^{-1}(j+1)} &= 1 + \frac{2(x_1 + \ldots + x_j)}{x_{n^{-1}(j+1)}} \\ &> 1 + \frac{2(x_1 + \ldots + x_{j-1} + x_{j+1})}{x_{n^{-1}(j+1)}} = r'_{n^{-1}(j+1)} \end{aligned} \tag{8}$$

and

$$r_{n^{-1}(j)} = 1 + \frac{2(x_1 + \ldots + x_{j-1})}{x_{n^{-1}(j)}} = r'_{n^{-1}(j)}. \tag{9}$$

No other competitive ratios change in the passage from $\mathcal{S}$ to $\mathcal{S}'$.

Case II: $x_{j+1} \leq x_{n^{-1}(j)}$. Form $\mathcal{S}'$ by interchanging $n(j)$ and $n(j + 1)$ (as before) and setting $n'(n^{-1}(j)) = j + 1$ and $n'(n^{-1}(j + 1)) = j$ (this is new.) We still have $r'_j = r_{j+1}$ and $r'_{j+1} = r_j$.

Now

$$r'_{n^{-1}(j+1)} = 1 + \frac{2(x_1 + \ldots + x_{j-1})}{x_{n^{-1}(j+1)}}; \tag{10}$$

this quantity is less than $r_{n^{-1}(j+1)}$. The last ratio we have to check is

$$r'_{n^{-1}(j)} = 1 + \frac{2(x_1 + \ldots + x_{j-1} + x_{j+1})}{x_{n^{-1}(j)}}. \tag{11}$$

Observe that $x_{n^{-1}(j+1)} < x_{n^{-1}(j)}$ (using the branch-increasing property of $\mathcal{S}$.) It follows that $r'_{n^{-1}(j)} < r_{n^{-1}(j+1)}$. As in case I, all other ratios are unchanged.

In either case, $\mathcal{S}'$ dominates $\mathcal{S}$ and we are done. □

4.2.2 Proof of Theorem 3.

The question that arises is the following: can one set forth an iterative scheme that converges to a well-defined limit in the space of strategies? Each iteration will of course be a single execution of the "simple interchange" primitive that we have just described. Since, *a priori*, an infinite number of interchanges may be required to render a search with a non-decreasing sequence of extents, some caution is required.

Observation. *For a progressive strategy, $x_k \to \infty$ as $k \to \infty$. (As usual, $\{x_k\} = \{x_k\}_{k=0}^{\infty}$ denotes the sequence of search extents.)*

Otherwise, there exists a value $M > 0$ for which $\{x_k\} \cap [1, M]$ is infinite; suppose this is so. For each branch b and each $M > 0$, the branch-b subsequence of $\{x_k\}$ has finitely many elements that are less than or equal to M, as it is an increasing and unbounded sequence. Since there are finitely many branches, we have derived a contradiction.

We need this most recent observation in order to define a limit strategy. Otherwise, we are vulnerable to the following pitfall: suppose we have a finite-ratio strategy and a value M for which $F := \{x_k\} \cap [1, M]$ is infinite. (Partly, the point we are making is that such strategies exist, although they cannot be branch-increasing.) After any finite number of the "simple interchanges" set out by Lemma 1, the finite-ratio property will persist. But in any non-decreasing-extent, "limiting" strategy, the extents in F would (all) have to precede those in $\{x_k\} \backslash F$. With F infinite, the finite-ratio property would clearly be destroyed. This is why we must have a progressive strategy before we start performing interchanges to bring the smaller search extents "to the front" of the sequence.

Algorithm. *Let $\{x_k\}$ be the extent sequence for a progressive strategy which fails to be non-decreasing. Initialize K to 0 and perform simple interchanges in the following fashion:*

1. *Select the smallest index m such that $x_{\min} := x_m = \min_{k=K}^{\infty}\{x_k\}$*
2. *Swap $x_{\min}$ and x_{m-1} etc. until $x_{\min}$ begins the new sequence $\{x'_k\}_{k=K}^{\infty}$.*

   At each interchange, modify the $n(\cdot)$ pointers as set out in the proof of Lemma 1, and check to see whether the no-repetition property has persisted. If not, there are two consecutive searches of one (or perhaps both) of the branches that were involved in the interchange. Where this occurs, delete the smaller of these two extents before proceeding. If $x_{\min}$ itself is deleted by this procedure, go to step 1 to reset $x_{\min}$.

3. *Set* $x_k \leftarrow x'_k$ *for all* $k \geq K$.

4. *Stop if* $\{x_k\}$ *is non-decreasing; otherwise, increment* K *and go to step 1.*

Suppose $\mathcal{S} = \mathcal{S}^{\text{init}}$ is a progressive strategy which fails to be non-decreasing; let $\mathcal{S}$ be fed as input to our algorithm. To show that this algorithm converges to a well-defined, progressive limiting strategy (hereafter denoted $\mathcal{S}^{\infty}$), we need to verify three things:

- for each k, the k^{th} search extent, x_k^{∞}, of the limiting strategy is well-defined
- the "next" sequence $n^{\infty}(\cdot)$ is well-defined, and $n^{\infty}(k) < \infty$ for all k
- each branch is visited at least once (for then the preceding item will imply that each branch is visited infinitely often.)

These three points suffice: it follows from the first two bulleted items that the k^{th} competitive ratio, r_k^{∞}, for the limiting strategy is well-defined (for all k). Moreover, the exposition will show that, for all k, r_k^{∞} appears in the ratio sequence for some intermediate strategy. This intermediate strategy is obtained from the initial strategy via finitely many simple interchanges and therefore dominates $\mathcal{S}$.

We use the index of the "big loop" (beginning in step 1 and falling through all the way to step 4) to keep track of intermediate strategies: let $\mathcal{S}^K$ be the strategy that is obtained upon completion of the index-K iteration of this loop. For all K, this strategy is progressive (the proof of Lemma 1 shows how to maintain the branch-increasing property; the no-repetition property is restored in step 2 of the algorithm wherever necessary; ergodicity is clearly maintained.) Note that each iteration of the big loop terminates, as it requires finitely many simple interchanges to complete. (Suppose for the sake of argument that we are in iteration $K = 0$ of the big loop. Then the set of extents which are smaller than x_0 is finite. Even if we delete $x_{\min}$ inside the subloop at step 2, this set loses a member. So we will eventually fall through to step 4. But there is nothing magic about $K = 0$.)

Let $\{x_k^j\}$ be the search extent sequence for $\mathcal{S}^j$; fix k_0. By construction, $x_{k_0}^j$ stops changing (as a function of the loop index j) for $j \geq k_0$. Said differently, the tail of the sequence $\{x_k^j\}_j$ is constant for each k (notice we are indexing on the "loop counter" now); let x_k^{∞} be this constant value.

We now show that $n^{\infty}(k)$ exists and is $< \infty$ for all k. This is a point of some subtlety. Suppose we have completed iteration k of the big loop in our algorithm (so x_k and all previous search extents have reached their asymptotic placement in the extent sequence. For readability, we have dropped the superscript from $x_k = x_k^\infty$.) We might attempt a direct proof that $n^\infty(k)$ is finite by looking at the set E of extents which

1. are smaller than $x_{n(k)}$, and
2. occur after $x_{n(k)}$ in the post-swap extent sequence.

It is not too hard to argue that E shrinks as the algorithm swaps its members ahead of $x_{n(k)}$ (this boils down to the fact that if $x'_{n'(k)} \neq x_{n(k)}$, the left-hand side is the smaller of the two values.) If (dropping the primes) $x_{n(k)}$ becomes the smallest extent not in its final sequence position, however, the algorithm will move it earlier in the extent sequence. In this process, the new pointer $n'(k)$ may get attached to a new, larger extent, repopulating the set E described earlier in this paragraph. How do we know that this process ever ends?

As before, let k be fixed. Let $U = U(k)$ be an integer greater than $\max\{(r_{\mathcal{S}} x_k^\infty - 3)/2, k\}$. Then $U > (r_{\mathcal{S}} x_k^U - 3)/2 = (r_{\mathcal{S}} x_k^\infty\text{-}3)/2$. Since each search extent is at least 1, we have trivially that

$$1 + 2(x_0^U + x_1^U + \ldots + x_U^U)/x_k^U > r_{\mathcal{S}}, \tag{12}$$

where the right-hand side is the worst-case competitive ratio for the initial strategy $\mathcal{S} = \mathcal{S}^{\text{init}}$.

Now $\mathcal{S}^U$ dominates $\mathcal{S}$. So r_k^U, the k^{th} competitive ratio for $\mathcal{S}^U$, is not greater than $r_{\mathcal{S}}$. But $\mathcal{S}^U$ is a progressive strategy, so we have the formula

$$r_k^U = 1 + 2(x_0^U + x_1^U + \ldots + x_{n^U(k)}^U)/x_k^U \leq r_{\mathcal{S}}. \tag{13}$$

This implies that $n^U(k) < U$. By construction, $x_{n^U(k)}^U = x_{n^U(k)}^\infty$. In words, once we reach strategy $\mathcal{S}^U$, all search extents up through the U^{th} (including the one in the previous equation) have stopped changing. As observed in the proof of Lemma 1, subsequent iterations of the "big loop" in the algorithm will not introduce any changes of branch assignment into this portion of the extent sequence. So $n^\infty(k)$ is well-defined (set it to $n^U(k) = n^{U+1}(k) = \ldots$).

Recall the notation that $\alpha(b)$ is the index of the first visit to branch b (for the initial strategy $\mathcal{S}$). Lastly, we need to make sure that each branch is visited, i.e. that $\alpha(b) < \infty$ for all b. Here we can get away with a crude estimate, again using the fact that each search extent is greater than or equal to 1. For integer $U > (r_{\mathcal{S}} - 3)/2$,

$$1 + 2(x_0^U + x_1^U + \ldots + x_U^U) > r_{\mathcal{S}} \geq r_{-1}^U$$

The definition of r_{-1}^U (as $\max_b(1 + 2(x_0^U + x_1^U + \ldots + x_{\alpha^U(b)}^U))$) shows that $\alpha^U(b) < U$ for all branches b and the rest of the argument is as before. This completes the proof of Theorem 3.

4.3 Solving for Optimal Strategies: The "Critical" Ratio

At the beginning of this section, we embarked on an effort to reduce the space of strategies that must be considered. It is worthwhile at this point to summarize and also to look ahead. We catalog a list of desirable properties in the following proposition. Most of the work has already been done to confirm that restricting to the corresponding subspace does not "throw out the baby with the bath water" (i.e. does not discard all of the optimal strategies.)

Proposition 1. *Without loss of generality, we can restrict our attention (in the unbounded case) to strategies $\mathcal{S}$ for which (all of) the following properties hold:*

1. *$\mathcal{S}$ is progressive (i.e., branch-increasing, non-repetitive and ergodic.)*
2. *The sequence $\{x_k\}$ of search extents is non-decreasing.*
3. *(Least-Searched) The branches are assigned in "least-searched-so-far" fashion. This says the following (for all k):*

$$\min_b \left[\max_{\{j | j<k \text{ and } b(j)=b\}} x_j \right]$$

   is assumed by branch $b = b(k)$.

4. *(cyclic) For any branch b, no other branch is explored more than once between sucessive visits to b. Up to renumbering, this says that the k^{th} search explores branch $k \pmod{m}$ for all k.*

Proof. The first two properties have already been substantiated. The least-searched property can be established by an interchange technique similar to that used for Theorem 3. The specifics are simpler than with the non-decreasing property just verified. This is because, if one goes to the first point where the property is violated and performs the obvious interchange of branch assignments, no difficulties are created. In particular, the extent sequence remains intact and the branch-increasing property is not destroyed.

The cyclic property is an easy consequence of the least-searched and non-decreasing properties. If the sequence of search extents is strictly increasing, it is obvious that the branch which was least recently searched must come next. (Our reckoning is in terms of the branch assignment sequence $\{b(k)\}$, as there is no notion of time in our problem formulation.) If the monotonicity of the extent sequence is not strict, the least-searched branch will not always be unique. But it is clear that we can break ties so as to preserve the cyclic property without violating the least-searched dictum.

Recall the no-repetition property: for all k, $b(k) \neq b(k+1)$. One further comment is required. Interchanges that are performed to establish the least-searched property may *a priori* defeat the no-repetition property. Not to worry, however: the latter gets restored when we go through the tie-breaker procedure to establish the cyclic property. This completes the proof of the proposition. □

Consider an exit placed at distance 1^+ from the source on branch $m-1$ (the last branch to receive its initial search. The meaning of the notation 1^+ is as described in the exposition following Definition 7: "just past" extent 1.) The searcher travels at least $2(m-1)+1^+$ total distance units to make the discovery. So no competitive ratio can be less than $2m-1$, even for the bounded problem. Analogous to the 2-branch problem, this ratio of $2m-1$ is achievable only if we know the *exact* distance of the exit t from the source s. Conversely, for any ratio strictly greater than $2m-1$, we can at least visit every branch once, i.e. we can choose ε small enough so that each branch can be searched to extent $1+\varepsilon$ without violating this ratio.

For a branch-increasing, cyclic search we can write the following constraint:

$$x_0 + \cdots + x_n \leq \rho x_{n-m+1}. \qquad (G_n)$$

Again we will refer to this inequality with n as a variable index, i.e. G_k means the same inequality with n replaced by k, and so on. Recall the notation that $\rho = (r-1)/2$, where r is some target ratio. (If $n < m-1$, set x_{n-m+1} to 1.) We reason as follows: suppose the exit is positioned so as to be discovered by search $n+1$. Given this information, the worst case is to be placed just past the previously searched extent on branch $b(n+1)$. Using the cyclic property, this corresponds to an exit location at distance x^+_{n-m+1} from the source. The total distance traveled to make the discovery is $2(x_0 + \cdots + x_n) + x^+_{n-m+1}$.

Observation. *Suppose we let $n > m$. Subtracting the equality form of (G_{n-1}) from that of (G_n) produces the formula $x_n = \rho x_{n-m+1} - \rho x_{n-m}$. The characteristic polynomial of this recurrence is $f_\rho(\xi) := \xi^m - \rho(\xi - 1)$.*

We pause to study this recurrence, which arises if we pursue a strategy of successive maximization as before. The recurrence has a solution $(x_j)_j$ for which $\lim_{j\to\infty} x_j = \infty$ iff f_ρ has a real root greater than 1. For, every complex number which is not a positive real has infinitely many powers with positive real part and infinitely many powers with negative real part. We claim that

$$\rho_1 > \rho_0 > 1, f_{\rho_0} \text{ has a real root } > 1 \Rightarrow f_{\rho_1} \text{ has a real root } > 1. \tag{14}$$

The claim is easily substantiated by verifying that

$$\text{for } \rho_1 > \rho_0 > 1, \xi > 1, \text{ we have } f_{\rho_1}(\xi) < f_{\rho_0}(\xi) \tag{15}$$

and observing that, for any real $\rho, f_\rho(\xi) \to \infty$ as $\xi \to \infty$.

To find the minimum value of ρ for which f_ρ has a real root > 1, we solve the equation $f_\rho(\xi) = 0$ for ρ, resulting in the formula

$$\rho = \xi^m/(\xi - 1). \tag{16}$$

As a function of ξ, the critical value is at $\xi = m/(m-1)$ (there is no other real critical value > 1.) The associated value of ρ is

$$\rho_{\text{crit}} := (m-1)\Big(\frac{m}{m-1}\Big)^m = m\Big(\frac{m}{m-1}\Big)^{m-1}.$$

It is easy to check that $m/(m-1) > 1$ is a double root of $f_{\rho_{\text{crit}}}$, and to confirm that $f_{\rho_{\text{crit}}}$ assumes its absolute minimum here (when thought of as a function with domain $(1, \infty)$.) Appealing again to Equation (15), we have that $f_\rho(\xi) > f_{\rho_{\text{crit}}} \geq 0$ whenever $\rho_{\text{crit}} > \rho > 1$ and $\xi > 1$. So ρ_{crit} really is the smallest value of the parameter ρ for which our polynomial has a real root greater than 1.

The roots of f_ρ vary continuously with ρ; as ρ dips below the critical value, the double root at $m/(m-1)$ bifurcates into a complex conjugate pair which we will call $\omega, \bar{\omega}$. The continuity property says that ρ can be chosen so that the argument of ω is arbitrarily close to zero. To summarize, as ρ decreases to subcritical values, we witness a transition: the recurrence $x_n = \rho x_{n-m+1} - \rho x_{n-m}$ no longer has a solution which approaches infinity in increasing fashion.

4.4 A Sequence of Mathematical Programs

Let x be the vector $(x_0, \ldots, x_n) \in R^{n+1}$, and let e_i denote the i^{th} standard basis vector. In vector notation, Inequality (G_n) becomes

$$x^T \cdot \big[(1, \ldots, 1)^T - \rho e_{n-m+1}\big] \leq 0. \tag{17}$$

When the subtraction is performed inside the brackets, note that the $1-\rho$ entry appears $m-1$ units to the left of the last (i.e. rightmost) 1. We will soon have occasion to work in R^k for $k > n+1$; in this situation we simply append the appropriate number of 0's (on the right) to the vector in brackets in (17).

Following the parallel with the 2-branch problem, we can describe compliance with the target ratio by constraints in a sequence of mathematical programs.

$$\begin{array}{rll} \max & \min & \{x_{k-m+2}, x_{k-m+3}, \ldots, x_k\} \\ \text{subj to} & B_k x & \leq \rho(e_0 + e_1 + \ldots + e_{m-2}) \\ & x_0 & \geq 1 \\ & x_i & \geq x_{i-1} \text{ for } i = 1, 2, \ldots, k. \end{array} \qquad (P_k)$$

Here is the motivation for the new objective function. Fix a strategy; let n^* be the smallest value of k for which $x_{k+1} < x_k$. For the sake of discussion, assume that the constraint upperbounding x_{n^*+1} is tight (and that $n^* < \infty$). This suggests that the $(n^*+1)^{st}$ search cannot be pursued to an extent which is large enough to be useful, while still returning to perform search n^*+2 in accordance with the target ratio. From the point of view of this worst-case ratio, we can do no better than to set $x_{n^*+1} = x_{n^*}$ and stop. If we had allowed the objective function to remain simply x_k, there would be $m-2$ other branches unaccounted for.

The vector of decision variables for (P_k) is $(x_0, \ldots, x_k) \in R^{k+1}$. Based on (17) and subsequent comments, the $(k+1)$x$(k+1)$ matrix B_k is lower triangular, with the value $1-\rho$ appearing throughout the $(m-1)^{st}$ subdiagonal. (In the 2-branch special case, this becomes the main subdiagonal.) Every other entry on or below the main diagonal is 1. In what follows, we number the rows and columns of B_k starting at 0 so as to be consistent with the indexing of the x_i's.

The right-hand side of the system $B_k x \leq \rho(e_0+e_1+\ldots+e_{m-2})$ expresses the "initial conditions" that $x_0 + \ldots + x_i \leq \rho$ for all $i \leq m-2$. To see that we have specified the correct number of initial conditions, note that the m^{th} search represents the first occasion on which we return to a branch (namely, branch 0) that has previously been explored. Recall that we constrain x_{m-1} by considering the worst case for discovery by search m. Thus the constraint for x_{m-1} marks the first appearance of the "general term", which has right-hand side 0.

4.5 Identifying a "Canonical" Solution

For the case $m = 2$, we showed in Section 2.2 that the "successive maximization" solution to the Program (P_k) is optimal whenever it is feasible. The argument given there also shows that, if this solution fails to be feasible, the Problem (P_k) itself is infeasible. We now encounter a new difficulty in our generalization to $m > 2$ branches: settling on a strategy of successive maximization no longer uniquely determines initial conditions for the resulting recurrence.

For general m, we will require that

$$x_0 + \cdots + x_{m-2} = \rho, \text{ and} \tag{18}$$

$$x_0 + \cdots + x_{m-2} + x_{m-1} = \rho x_0. \tag{19}$$

It is clear that $x_0 + \cdots + x_{m-2} \leq \rho$ is the first constraint from our sequence of mathematical programs that can be tight, since we want all search extents to be strictly positive. Of course, this pair of equations is underdetermined for $m > 2$; we describe a way to identify a choice of $x_0, x_1, \ldots, x_{m-1}$ with desirable properties.

At $\rho = \rho_{\text{crit}}$, we set

$$x_k = \big(C + (k+1)D\big)\Big(\frac{m}{m-1}\Big)^{k+1}. \tag{20}$$

All we are doing is writing down that portion of the general solution (to the recurrence $x_k = \rho(x_{k-1} - x_{k-m})$) corresponding to the double root at $m/(m-1)$. Equations (18) and (19) now uniquely determine C and D. Subtracting Equation (18) from Equation (19) and using the fact that $m/(m-1)$ is a characteristic root of the recurrence leads easily to $C = 1$. It is then clear that D is a strictly positive real (by plugging in values of ρ_{crit} and of the x_is to Equation (18), say). Therefore $\{x_k\}$ is a strictly increasing sequence of positive numbers.

For $\rho < \rho_{\text{crit}}$, let $\omega, \bar{\omega}$ be the complex pair of characteristic values which are "near" $m/(m-1)$. For definiteness, let ω be the member of this pair with positive real part. Set

$$x_k = \alpha\omega^{k+1} + \bar{\alpha}\bar{\omega}^{k+1}. \tag{21}$$

As before, we have two unknowns (the real and imaginary parts of α) that are uniquely determined by Equations (18) and (19). It is easy to show that re$(\alpha) = 1/2$. Thus x_k is of the form $\frac{1}{2}\Big[(1+\delta i)\omega^{k+1}+(1-\delta i)\bar{\omega}^{k+1}\Big]$; Equation (18) now determines the real number δ.

We claim that the value of x_0 thus specified (and in fact that of each subsequent element of the extent sequence $\{x_k\}$) is continuous as a function of ρ. In particular, this is true as $\rho \nearrow \rho_{\text{crit}}$. A corollary is that the maximum searchable extent approaches infinity as $\rho \nearrow \rho_{\text{crit}}$. This is a non-trivial conclusion, as we will show that this extent cannot be infinite for subcritical ρ.

Here is how the continuity claim can be verified. It is trivial to check that $x_k = (\omega + \bar{\omega})x_{k-1} - \omega\bar{\omega}x_{k-2}$ whenever $k \geq 1$, with the proviso that x_{-1} is just 1. So x_1 is a function of x_0 and ω which is linear in x_0. In turn, x_2 is a linear function of x_1, and so on. Since linearity is preserved under composition and sums, $x_0 + \cdots + x_{m-2}$ can be written in the form $g(x_0, \omega)$, where g is linear in its first argument. It is a simple matter to verify that the "coefficient" of x_0 in this function is non-zero. Furthermore, none of the formulas change when we work at the critical value of ρ, replacing ω by $m/(m-1)$. In either case, Equation (18) becomes $g(x_0, \omega) = \rho$. It is trivial to solve for x_0, yielding a formula that is continuous in ρ and in ω; the latter, being a root of a polynomial parameterized by ρ, is continuous as a function of ρ as well. As a result, we have the following conclusion.

Proposition 2. *Let the sequence $\{x_k^*\}$ be determined by Equations (18), (19) and (21) for subcritical ρ (alternatively, Equations (18), (19) and (20) at the critical value of ρ.)*

Then for all k, x_k^* is continuous as a function of ρ as $\rho \nearrow \rho_{crit}$. Thus for fixed K, $\{x_k^*\}_{k=0}^K \in R^{k+1}$ is continuous as a function of ρ. We therefore have that, for any fixed K, $\{x_k^*\}_{k=0}^K$ is increasing whenever ρ is sufficiently close to its critical value.

However, this convergence is far from uniform: since $\arg(\omega)$ *is non-zero for fixed sub-critical ρ, we have that $x_k^* < 0$ for some k.*

One more comment is in order regarding the content of this proposition: referring back to Equations (20) and (21) and using what we know so far, one can see that $\delta := \text{im}(\alpha)$ must approach $-\infty$ as ρ approaches its critical value (for $\text{im}(\omega)$ approaches 0 simultaneously). This is easier to obtain in the 2-branch case, where we can exhibit explicit formulas for ω and the constants D and δ from the aforementioned equations. Here we have to work a little harder; the foregoing proposition tells us that the analogy with the 2-branch case is in fact robust.

4.6 Infeasibility for Subcritical Ratios

Theorem 4. *Let m be the number of branches. For any $\rho < \rho_{crit} = m\left(\frac{m}{m-1}\right)^{m-1}$, there exists k for which the Program (P_k) of Section 4.4 is infeasible. Thus $2\rho+1$ is not a feasible competitive ratio for the unbounded problem.*

Further, as $\rho \nearrow \rho_{crit}$, the searchable extent under competitive ratio $2\rho+1$ increases to infinity.

The claim about the searchable extent increasing to ∞ as $\rho \nearrow \rho_{\text{crit}}$ was demonstrated in the course of the previous section.

The rest of this section will be devoted to the proof of the infeasibility claim for $\rho < \rho_{\text{crit}}$. We will refer to the following relaxation of the Program (P_k) as the "primal" problem. Here we rewrite the constraint $x_0 \geq 1$ in vector form and replace constraints of the form $x_{i-1} \leq x_i$ by nonnegativity constraints.

$$\begin{array}{rrl} \max & \min & \{x_{k-m+2}, x_{k-m+3}, \ldots, x_k\} \\ \text{subj to} & B_k x & \leq \rho(e_0 + e_1 + \ldots + e_{m-2}) \\ & -e_0^T x & \leq -1 \\ & x & \geq 0 \end{array} \qquad (Q_k)$$

The problem (Q_k) is readily converted to a linear program (Q'_k) by introducing a new variable z and the $m-1$ constraints $z \leq x_{k-m+i}, i = 2, \ldots, m$. Then the objective function shown is replaced by z itself. As a result, it is easy to see that (Q'_k) is feasible if and only if (Q_k) is feasible. So we concentrate on the feasible region of the latter for simplicity.

Recall that the $(k+1)$x$(k+1)$ matrix B_k is lower triangular, with the value $1-\rho$ appearing throughout the $(m-1)^{st}$ subdiagonal. In what follows, we number the rows and columns of B_k starting at 0 so as to be consistent with the indexing of the x_i's.

Farkas' lemma says that the Problem (Q_k) is feasible iff the following system is infeasible:

$$\begin{array}{rl} [B_k^T | -e_0] \begin{bmatrix} y \\ v \end{bmatrix} & \geq 0 \\ y & \geq 0 \\ v & \geq 0 \\ [\rho(e_0^T + \ldots + e_{m-2}^T) | -1] \begin{bmatrix} y \\ v \end{bmatrix} & < 0. \end{array} \qquad (FD_k)$$

Here $y \in R^{k+1}$ and v is a scalar. The last constraint boils down to $\rho(y_0 + \cdots + y_{m-2}) < v$.

The main content of this section follows from a rather simple observation. Let us ignore v for the moment (or set it to zero) and look at the system $B_k^T y \geq 0$. Now B_k^T is upper triangular, with $(1-\rho)$'s along the $(m-1)^{st}$ *super*diagonal. Therefore the n^{th} constraint is, after a little algebra,

$$y_n + y_{n+1} + ... + y_k \geq \rho y_{n+m-1}. \tag{22}$$

Observation. *Inserting equality constraints in the system $B_k^T y \geq 0$ gives rise to the* same *recurrence as in the "primal" system $B_k x \leq \rho(e_0 + \ldots + e_{m-2})$. The only difference is that the subscripts run backwards, i.e. we now have $y_n = \rho(y_{n+m-1} - y_{n+m})$.*

The next several paragraphs are technical and, in particular, the subscripts become confusing. To build intuition, the reader may want to go through the argument once with $m = 3$ and then return to general $m > 2$. To show that $2\rho + 1$ is an infeasible competitive ratio for the unbounded problem, we want to exhibit a solution of the Farkas dual problem (FD_k) for appropriately chosen k. The particulars are as follows: let $\eta = \eta(\rho)$ be the largest index K for which $\{x_k^*\}_{k=0}^K$ is nondecreasing. Here x_k^* is the generic term in the "canonical primal solution" of Proposition 2. By definition of η we have that $x_{\eta+m}^* = \rho(x_{\eta+1}^* - x_\eta^*)$ is the first negative term of the sequence $\{x_k^*\}$.

For ρ sufficiently close to its critical value, we know that $\eta > m$ and that $x_{\eta+m+1}^*, \ldots, x_{\eta+2m-2}^*$ are all negative (because $\arg(\omega)$ is close to 0). The recurrence says in turn that $x_\eta^* > x_{\eta+1}^* > \cdots > x_{\eta+m-1}^*$. Furthermore, there is no loss of generality in assuming that ρ is close to its critical value, since the feasible region for Problem (P_k) shrinks as ρ decreases, and we are trying to prove infeasibility for large k whenever $\rho < \rho_{\text{crit}}$.

We work with $(FD_{\eta+2m-2})$, setting $y_{\eta+2m-2} = x_0^*$, $y_{\eta+2m-3} = x_1^*$, ..., $y_{m-1} = x_{\eta+m-1}^*$. Let $y_{m-2} = y_{m-3} = \cdots = y_0 = 0$. The non-negativity constraints are satisfied by these choices of the y_ks; in view of this, constraints $\eta+2m-2, \eta+2m-3, \ldots \eta+m$ from the system $B^T y - e_0 v \geq 0$ are redundant. For readability, we have dropped the subscript from the matrix B^T.

Constraint $\eta + m - 1$ from this system is

$$y_{\eta+m-1} + y_{\eta+m} + \cdots + y_{\eta+2m-2} \geq \rho y_{\eta+2m-2}. \tag{23}$$

By construction, this is satisfied as an equality: it restates the fact, established in the previous section, that $x_0^* + x_1^* + \cdots + x_{m-1}^* = \rho x_0^*$.

It now becomes clear that constraints $\eta + m - 1, \eta + m - 2, \ldots, m - 1$ from the system $B^T x - e_0 v \geq 0$ are all satisfied as equalities; we are just "riding the recurrence". For, $y_{\eta+m-2} + y_{\eta+m-1} + \cdots + y_{\eta+2m-2} = \rho y_{\eta+2m-3}$ iff $y_{\eta+m-2} = \rho(y_{\eta+2m-3} - y_{\eta+2m-2})$ via the observation of the previous paragraph; the second of these equations is of course true. Now continue with constraint $\eta + m - 3$ and so on inductively.

It remains to check constraints $m - 2, m - 3, \ldots 1$ and then to assign a value to the variable v for which constraint 0 also holds. After filling in 0 for y_{m-2}, constraint $m - 2$ becomes

$$y_{m-1} + y_m + \ldots + y_{\eta+2m-2} \geq \rho y_{2m-3}. \tag{24}$$

The LHS of this inequality is $\rho y_{2m-2} = \rho x^*_\eta$ as established in the previous paragraph. The RHS is $\rho x^*_{\eta+1}$; by definition of η we have $\rho x^*_\eta > \rho x^*_{\eta+1}$ as desired.

Since $y_{m-3}, y_{m-4}, \ldots, y_1$ are all zero, ρx^*_η is the LHS for *each* of the constraints $m - 3, m - 4, \ldots 1$. The right-hand sides are $\rho y_{2m-4} = \rho x^*_{\eta+2}$, $\rho y_{2m-5} = \rho x^*_{\eta+3}$, …, $\rho y_m = \rho x^*_{\eta+m-2}$ respectively. In light of the earlier observation that the x^*s are decreasing from subscript $\eta+1$ through subscript $\eta + m - 1$, we see that constraints $m - 3, m - 4, \ldots, 1$ are all satisfied.

Lastly, constraint 0 boils down to $\rho x^*_\eta - v \geq \rho x^*_{\eta+m-1}$. We assign to v the value $\rho(x^*_\eta - x^*_{\eta+m-1}) > 0$. Thus the system of inequalities $B^T y - e_0 v \geq 0$ is verified, as is the non-negativity constraint on v. We also have $\rho(y_0 + y_1 + \cdots + y_{m-2}) - v = 0 - \rho(x^*_\eta - x^*_{\eta+m-1}) < 0$. So the Problem $(FD_{\eta+2m-2})$ is feasible. This finishes the proof that the "primal" problem $(Q_{\eta+2m-2})$ is infeasible. Therefore the original problem $(P_{\eta+2m-2})$ is infeasible and Theorem 4 is proved.

5 Concluding remarks

Under various deterministic and probabilistic information we have been able to find strategies with proven optimal competitive ratio. To the best of our knowledge, our results represent the first rigorous quantitative analysis of the value of additional information (deterministic or probabilistic) for on-line search problems. We have mainly considered the m-concurrent branch problem, but the qualitative implications of our findings will apply to more general search. In order to give another perspective on the results, it is interesting to note that, for the 2-concurrent branch problem, the information "the exit is within 233 units of the origin" is about equivalent (in the sense that it leads to optimal strategies with same competitive ratios) to the information "the probability that the exit is on branch 1 is 0.25". More generally the correspondence is as follows:

Equivalence between probabilistic and deterministic information		
Probability	Distance	Corresponding Ratio
0.5	∞	9
0.45	4.3×10^{15}	8.973
0.4	2.8×10^{7}	8.892
0.35	4.7×10^{5}	8.754
0.3	1,755.81	8.554
0.25	232.93	8.284
0.2	56.42	7.929
0.15	18.30	7.465
0.1	7.87	6.844
0.05	3.62	5.937
0.01	1.78	4.562

One can see that in order to substantially improve the unbounded case (with a ratio of 9), one has to receive very restrictive information.

Acknowledgements:
We would like to thank two anonymous referees and Suvrajeet Sen for comments that greatly helped improving the clarity of the paper.

References

[1] S. Alberts. Competitive online algorithms, overview. *Optima, Mathematical Programming Society Newsletter*, 54:1–8, 1997.

[2] S. Ascheuer, M. Gröetschel, J. Rambau, and N. Kamin. Combinatorial online optimization in practice. *Optima, Mathematical Programming Society Newsletter*, 57:1–6, 1998.

[3] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106:234–244, 1993.

[4] R. Bellman. A minimization problem. *Bulletin of the American Mathematical Society*, 62:270, 1956.

[5] A. Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[6] A. Borodin, N. Linial, and M. Saks. An optimal on-line algorithm for metrical task systems. *Journal of the ACM*, 39:745–763, 1992.

[7] D.P. Foster and R. Vohra. Regret in the on-line decision problem. Working paper, Department of Management Science, Ohio State University, 1997.

[8] B. Gluss. An alternative solution to the 'lost at sea' problem. *Naval Research Logistics Quaterly*, 8:117–128, 1961.

[9] B. Gluss. The minimax path in a search for a circle in the plane. *Naval Research Logistics Quaterly*, 8:357–360, 1961.

[10] R. Hassin and A. Tamir. Minimal length curves that are not embeddable in an open planar set: The problem of a lost swimmer with a compass. *SIAM Journal of Control and Optimization*, 30:695–703, 1992.

[11] D. Heyman and M. Sobel. *Stochastic Models in Operations Research*, volume 2. Mc Graw-Hill, 1984.

[12] J. Isbell. An optimal search pattern. *Naval Research Logistics Quaterly*, 4:357–359, 1957.

[13] P. Jaillet and M. Stafford. Parallel on-line search. Working paper, MSIS Department, The University of Texas at Austin, December 1997.

[14] M. Kao, J. Reif, and S. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. In *Proc. of the 4th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 441–447, 1993.

[15] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator. Competitive snoopy caching. *Algorithmica*, 3:79–119, 1988.

[16] M. Manasse, L. McGeoch, and D. Sleator. Competitive algorithms for on-line problems. *Journal of Algorithms*, 11:208–230, 1990.

[17] C. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84:127–150, 1991.

[18] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.