Online Search with Maximum Clearance

Spyros Angelopoulos, [1, 3] **Malachi Voss**[2, 3]
[1] Centre National de la Recherche Scientifique (CNRS)
[2] École Normale Supérieure
[3] Sorbonne Université, Laboratoire d'informatique de Paris 6, LIP6, F-75252 Paris, France.
spyros.angelopoulos@lip6.fr, malachi.voss@ens.fr

Abstract

We study the setting in which a mobile agent must locate a hidden target in a bounded or unbounded *environment*, with no information about the hider's position. In particular, we consider *online* search, in which the performance of the search strategy is evaluated by its worst case *competitive ratio*. We introduce a multi-criteria search problem in which the searcher has a *budget* on its allotted search time, and the objective is to design strategies that are competitively efficient, respect the budget, and maximize the total searched ground. We give analytically optimal strategies for the line and the star environments, and efficient heuristics for general networks.

Introduction

We study a general search problem, in which a mobile agent with unit speed seeks to locate a target that hides in some unknown position of the environment. Specifically, we are given an environment which may be bounded or unbounded, with a point O designated as its *root*. There is an immobile *target* (or *hider*) H that is hiding in some unknown point in the environment, whereas the searcher is initially placed at the root O. The searcher has no information concerning the hider's position. A *search strategy* S determines the precise way in which the searcher explores the environment, and we assume deterministic strategies. The *cost* of S given hider H, denoted by $d(S, H)$, is the total distance traversed by the searcher the first time it reaches the location of H, or equivalently the total search time.

There is a natural way to evaluate the performance of the search strategy that goes back to (Bellman 1963) and (Beck and Newman 1970): we can compare the cost paid by the searcher in a worst-case scenario to the cost paid in the ideal situation where the searcher knows the hider's position. We define the *competitive ratio* of strategy S as

$$\texttt{cr}(S) = \sup_H \frac{d(S,H)}{d(H)}, \tag{1}$$

with $d(H)$ the distance of H from O in the environment.

Competitive analysis allows to evaluate a search strategy under a status of complete uncertainty, and provides strict, worst-case guarantees. Competitive analysis has been applied to several search problems in robotics, for example (Sung and Tokekar 2019), (Magid and Rivlin 2004), (Taylor and Kriegman 1998) (Isler, Kannan, and Daniilidis 2003). See also the survey (Ghosh and Klein 2010).

In this work we will study the following classes of environments: First, we consider the problem of searching *on the line*, informally known as the *cow path* problem (Kao and Littman 1997), in which the environment is the unbounded, infinite line. Next, we consider a generalization of linear search, in which the environment consists of m unbounded *rays*, concurrent at O; this problem is known as the m*-ray search* or *star search* problem. This environment can model much broader settings in which we seek an intelligent allocation of resources to tasks under uncertainty. Thus, it is a very useful paradigm that arises often in applications such as the design of *interruptible* systems based on *contract* algorithms (Bernstein, Finkelstein, and Zilberstein 2003; Angelopoulos 2015; Kupavskii and Welzl 2018), or *pipeline filter ordering* (Condon et al. 2009). Last, we consider general undirected, edge-weighted graph *networks*, and a target that can hide anywhere over an edge or a vertex of this graph.

In some previous work, online search may refer to the setting in which the searcher has no information about the environment or the position of the target. In this work we assume that the searcher knows the environment, but not the precise position of the target. This is in line with some foundational work on competitive analysis of online search algorithms, e.g. (Koutsoupias, Papadimitriou, and Yannakakis 1996).

Searching With a Budget

Most previous work on competitive analysis of searching has assumed that a target is indeed present, and so the searcher will eventually locate it. Thus, the only consideration is minimizing the competitive ratio. However, this assumption does not reflect realistic settings. Consider the example of Search-And-Rescue (SAR) operations: first, it is possible that the search mission may fail to locate the missing person, in which case searching should resume from its starting point instead of continuing fruitlessly for an exorbitant amount of time. If, say, the first day's efforts were unsuccessful, it would be best to have covered as much ground as possible before restarting. Second, and more importantly, SAR operations come with logistical constraints and limited resources, notably in terms of the time alloted to the mission.

To account for such situations, in this work we study online search in the setting where the searcher has a certain *budget* T, which reflects the total amount of search time that it can afford, and a desired competitive ratio R that the search must attain. If the target is found within this budget, the search is *successful*, otherwise it is deemed *unsuccessful*. We impose two optimization constraints on the search. First, it must be competitively efficient, i.e., its competitive ratio, as expressed by (1) is at most R, whether it succeeds or not. Second, if the search is unsuccessful, the search has maximized the total *clearance* by time T. In the case of the environments we study in this work, the clearance is the measure of the part of the environment, i.e, the total length, that the searcher has explored by time T. We call this problem the *Maximum Clearance problem with budget T and competitive ratio R*, and we denote it by MAXCLEAR(R,T).

It should be clear that the competitive ratio and the clearance are in a trade-off relation with respect to any given budget T: by reducing the competitive efficiency, one can improve the clearance, and vice versa. Hence, our goal is to find strategies that attain the optimal tradeoff, in a Pareto sense, between these two objectives.

Contributions

We study Maximum Clearance in three environments: the unbounded line, the unbounded star, and a fixed network. We begin with the line: here we show how to use a linear programming formulation to obtain a Pareto-optimal solution. We also show that the Pareto-optimal strategy has a natural interpretation as the best among two simple strategies.

We then move to the m-ray star, which generalizes the line, and which is more challenging. Here, we argue that the intuitive strategies that are optimal for the line are not optimal for the star. We thus need to exploit the structure of the LP formulation, so as to give a Pareto-optimal strategy. We do not require an LP solver, instead, we show how to compute the theoretically optimal strategy efficiently, in time $O(m \log m \log T + m \log T \log \log T)$. Experimental evaluations confirm the superiority of this optimal strategy over other candidate solutions to the problem.

Finally, we consider the setting in which the environment consists of a network. Here, there is a complication: we do not known the optimal competitive ratio as, for example, in the star (the problem is NP-hard if the target hides on vertices), and only $O(1)$ approximations of the optimal competitive ratio are known (Angelopoulos and Lidbetter 2020). Hence, in this context, we define MAXCLEAR(R,T) with $R \geq 1$, as the problem of maximizing clearance given budget T, while guaranteeing that the strategy is an *R-approximation* of the optimal competitive ratio. Previous approaches to competitive searching in networks typically involve a combination of a solution to the Chinese Postman Problem (CPP) (Edmonds and Johnson 1973) with iterative doubling of the search radius. For our problem, we strengthen this heuristic using the *Rural Postman Problem* (RPP) (Frederickson, Hecht, and Kim 1978), in which only a subset of the network edges need to be traversed. While RPP has been applied to the problem of online coverage in robotics (Xu and Stentz 2010), (Easton and Burdick 2005), to the best of our knowledge, no previous work on competitive search has addressed its benefits. Although there is no gain on the theoretical competitive ratio, our experimental analysis shows that it has significant benefits over the CPP-based approach. We demonstrate this with experiments using real-world data from the library *Transportation Network Test Problems* (Bar-Gera 2002), which model big cities.

We conclude with some extensions and applications. We first explain how our techniques can be applied to a problem "dual" to Maximum Clearance, which we call *Earliest Clearance*. We also show some implications of our work for *contract scheduling* problems. In particular, we explain how our results extend those of (Angelopoulos and Jin 2019) for contract scheduling with end guarantees.

Due to space limitations, we omit several technical proofs. We refer the reader to (Angelopoulos and Voss 2020) for the full version of this paper.

Other Related Work

It has long been known that linear search has optimal competitive ratio 9 (Beck and Newman 1970), which is achieved by a simple strategy based on iterative doubling. Star search on m rays also has a long history of research, going back to (Gal 1974) who showed that the optimal competitive ratio is

$$R^*_m = 1 + 2\rho^*_m, \text{ where } \rho^*_m = \frac{m^m}{(m-1)^{m-1}}, \qquad (2)$$

a result that was later rediscovered by computer scientists (Baeza-Yates, Culberson, and Rawlins 1993). Star search has been studied from the algorithmic point of view in several settings, such as randomized strategies (Kao, Reif, and Tate 1996); multi-searcher strategies (López-Ortiz and Schuierer 2004); searching with an upper bound on the target distance (Hipke et al. 1999; Bose, Carufel, and Durocher 2015); fault-tolerant search (Kupavskii and Welzl 2018); and probabilistic search (Jaillet and Stafford 1993; Kao and Littman 1997). For general, edge-weighted networks only $O(1)$-approximation strategies are known (Koutsoupias, Papadimitriou, and Yannakakis 1996; Angelopoulos and Lidbetter 2020).

Preliminaries

For the m-ray star, we assume the rays are numbered $0, \ldots, m-1$. A search strategy for the star is defined as $\{(x_i, r_i)\}_{i \geq 1}$, with the semantics that in the i-th *step*, the searcher starts from O, visits ray r_i to length x_i, then returns to O. A *cyclic* strategy is a strategy for which $r_i = i \bmod m$; we will thus often omit the r_i's for such strategies, since they are implied. We make the standing assumption that the target is hiding at least at unit distance from the root, otherwise there is no strategy of bounded competitive ratio.

A *geometric* strategy is a cyclic strategy in which $x_i = b^i$, for some $b > 1$, which we call the *base*. Geometric strategies are important since they often give optimally competitive solutions to search problems on a star. For instance, the optimal competitive ratio R^*_m is achieved by a geometric strategy with base $b = \frac{m}{m-1}$ (Gal 1974). In general, the competitive ratio of a cyclic strategy with base b is equal to $1 + 2\frac{b^m}{b-1}$ (Gal 1972). By applying standard calculus, it follows that, for any given $R = 1 + 2\rho \geq R^*_m$, the geometric strategy with base b is R-competitive if and only if $b \in [\zeta_1, \zeta_2]$, where ζ_i are the positive roots of the characteristic polynomial $p(t) = t^m - \rho t + \rho$.

A less known family of strategies for the m-ray star is the set of strategies which maximize the searched length at the i-th step. Formally, we want x_i to be as large as possible, so that the strategy $X = (x_i)$ has competitive ratio $R = 1+2\rho$. It turns out that this problem has indeed a solution, and as shown in (Jaillet and Stafford 1993), the resulting strategy $Z = (z_i)$ is one in which the search lengths are defined by the linear recurrence relation $z_{m+i} = \rho(z_{i+1} - z_i)$. (Jaillet and Stafford 1993) give a solution to the recurrence for $\rho = \rho^*_m$. We can show that Z is in fact uniquely defined for all values of $R \geq R^*_m$, and give a closed-form expression for z_i, as a function of ζ_1 and ζ_2, defined above. Following the terminology of (Angelopoulos, Dürr, and Jin 2019) we call Z the *aggressive strategy of competitive ratio* R, or simply the *aggressive* strategy when R is implied.

For the star we will use a family of linear inequalities involving the search lengths x_i to model the requirement that the search is R-competitive. Such inequalities are often used in competitive search, see e.g. (López-Ortiz and Schuierer 2001), (Hipke et al. 1999). Each inequality comes from an adversarial position of the target: for a search strategy of the form $X = \{(x_i, r_i)\}$ in the star, the placements of the target which maximize the competitive ratio are on ray r_j and at distance $x_j + \epsilon$, for all j and for infinitesimally small ϵ (i.e., the searcher barely misses the target at step j).

There is, however, a subtlety in enforcing competitiveness in our problem. In particular, we need to filter out some strategies that can be R-competitive up to time T, but are artificial. To illustrate this, consider the case of the line, and a strategy S that walks only to the right of O up to time T (it helps to think of T as very large). This strategy is 1-competitive in the time interval $[0, T]$, and obviously maximizes clearance, but intuitively is not a realistic solution. The reason for this is that S discards the entire left side with respect to R-competitiveness. Specifically, for a point at distance 1 to the left of O, any extension S' of S will incur a competitive ratio of at least $2T + 1$, which can be enormous.

We thus need to enforce a property that intuitively states that a feasible strategy S to our problem should be extendable to an R-competitive strategy S' that can detect targets hiding infinitesimally beyond the boundary that has been explored by time T in S. We call this property *extendability* of an R-competitive strategy (see the full version (Angelopoulos and Voss 2020) for a detailed discussion). Our experimental evaluation shows that the optimal extendable strategy on the star performs significantly better than other candidate strategies, which further justifies the use of this notion.

A Warm-up: Maximum Clearance on the Line

We begin with the simplest environment: an unbounded line with root O. Fix a competitive ratio $R = 1 + 2\rho$, for some $\rho \geq \rho^*_2 = 4$. Without loss of generality, we assume cyclic strategies $X = (x_i)$ such that $x_{i+2} > x_i$, for all i.

Let $\mathcal{S}_k$ denote the set of all strategies $X = (x_1, \ldots x_k)$ with k steps. We can formulate MAXCLEAR(R,T) *restricted* to $\mathcal{S}_k$ using the following LP, which we denote $L_2^{(k)}$.

$$\begin{aligned}
\max \quad & x_{k-1} + x_k && (L_2^{(k)})\\
\text{subject to} \quad & x_1 \leq \rho && (C_0)\\
& \sum_{i=1}^{j+1} x_i \leq \rho \cdot x_j, \qquad j \in [1, k-2] && (C_j)\\
& \sum_{i=1}^{k} x_i \leq \rho \cdot x_{k-1} && (E_{k-1})\\
& 2\sum_{i=1}^{k-1} x_i + x_k \leq T && (B)
\end{aligned}$$

In this LP, constraints (C_0) and $(C_1), \ldots (C_{k-2})$ model the requirement for $(1 + 2\rho)$-competitiveness. (C_0) models a target hiding at distance 1 from O, whereas the remaining constraints model a target hiding right after the turn points of $x_1, \ldots x_{k-2}$, respectively. Constraint (B) is the budget constraint. Last, constraint (E_{k-1}) models the extendability property, which on the line means remaining competitive for a target hiding just beyond the turn point of x_{k-1}. For more details on this type of constraints, see the discussion for the more general m-ray star problem.

Therefore, an optimal strategy is one of maximum objective value, among all feasible solutions to $L_2^{(k)}$, for all $k \geq 1$. We will use this formulation to show that the optimal strategy has an intuitive statement. Let $Z = (z_i)$ be the aggressive strategy of competitive ratio R. From Z we derive the *aggressive strategy with budget* T, which is simply the maximal prefix of Z that satisfies the budget constraint (B). We denote this strategy by Z_T.

Note that Z_T may be wasteful, leaving a large portion of the budget unused, which suggests another intuitive strategy derived from Z. Informally, one can "shrink" the search lengths of Z in order to deplete the budget precisely at some turn point. Formally, we define the *scaled aggressive strategy with budget* T, denoted by $\tilde{Z}_T$ as follows. Let l be the minimum index such that $2\sum_{i=1}^{l-1} z_i + z_l \geq T$, and define γ as $T/(2\sum_{i=1}^{l-1} z_i + z_l)$. Then $\tilde{Z}_T$ is defined as $(\tilde{z}_i) = (\gamma \cdot z_i)$.

We will prove that one of Z_T, and $\tilde{Z}_T$ is the optimal strategy. We can first argue about constraint tightness in an optimal solution to $L_2^{(k)}$.

Lemma 1. *In any optimal solution to $L_2^{(k)}$, at least one of the constraints (C_0) and (B) is tight, and all other constraints must be tight.*

Lemma 1 shows that if X^* is optimal for $L_2^{(k)}$, then one can subtract successive constraints from each other to obtain the linear recurrence relation $x^*_{i+2} = \rho(x^*_{i+1} - x^*_i)$, with constraint (C_1) giving an initial condition. So X^*, viewed as a point in $\mathbb{R}^k$, is on a line $\Delta \subset \mathbb{R}^k$, defined as the set of all points which satisfy $(C_1), \ldots, (E_{k-1})$ with equality. This leaves us with two possibilities: either $X^* = X_0^{(k)}$ the point on Δ for which (C_0) is tight, or $X^* = X_B^{(k)}$ the point on Δ for which (B) is tight.

Define now $\mathcal{X}_0$ as the set of all feasible points $X_0^{(k)}$ and $\mathcal{X}_B$ as the set of all feasible points $X_B^{(k)}$. A point X is *optimal* for one of these sets if its objective value is no worse than any point in that set. The following lemma is easy to see for Z_T, and requires a little more effort for $\tilde{Z}_T$.

Lemma 2. *Z_T is optimal for $\mathcal{X}_0$, and $\tilde{Z}_T$ is optimal for $\mathcal{X}_B$.*

From Lemma 1 and 2 we conclude that the better of the two strategies Z_T and $\tilde{Z}_T$ is optimal for MAX(R,T) on the line. We call this strategy the *mixed aggressive* strategy.

Maximum Clearance on the Star

We now move to the m-ray star domain. We require that the strategy be $(1 + 2\rho)$-competitive, for some given $\rho \geq \rho_m^*$, where $\rho_m^* = \frac{m^m}{(m-1)^{m-1}}$, and we are given a time budget T.

A First, but Suboptimal Approach

An obvious first place to look is the space of geometric strategies. We wish the geometric strategy to have competitive ratio $1 + 2\rho$, so the strategy must have base $b \in [\zeta_1, \zeta_2]$, using the notation of the preliminaries. Since we want to maximize the clearance of our strategy, it makes sense to take $b = \zeta_2$. We define the *scaled geometric strategy with budget T* similarly to the scaled aggressive strategy: find the first step at which the budget T is depleted, and scale down the geometric strategy so that it depletes T precisely at the end of that step. The scaled geometric strategy represents the best known strategy prior to this work, but is suboptimal.

For Maximum Clearance on the line, we proved that the optimal strategy is the best of the aggressive and the scaled aggressive strategies. One may ask then whether the optimal strategy in the star domain can also be expressed simply as the better of these two strategies. The answer is negative, as we show in the experimental evaluation.

Modeling as an LP

As with the line, we first show how to formulate the problem using a family of LPs, denoted by L_k^m, partitioning strategies according to their length k. For a given step j, we denote by $\bar{\jmath}$ the previous step for which the searcher visited the same ray, i.e, the maximum $\bar{\jmath} < j$ such that $r_{\bar{\jmath}} = r_j$, assuming it exists, otherwise we set $x_{\bar{\jmath}} = 1$. We denote by l_r the last step at which the searcher explores ray r. Finally, we denote by j_0 the last step in which the searcher searches a yet unexplored ray, i.e., the largest step j such that $\bar{\jmath} = 0$. This gives us:

$$
\begin{aligned}
&\max && \sum\nolimits_{i=1}^{m} x_{l_i} && (L_m^{(k)})\\
&\text{subject to} && \sum\nolimits_{i=1}^{j_0} x_i \leq \rho && (C_0)\\
& && \sum\nolimits_{i=1}^{j-1} x_i \leq \rho \cdot x_{\bar{\jmath}}, \qquad j \in [j_0+1, k] && (C_j)\\
& && \sum\nolimits_{i=1}^{k} x_i \leq \rho \cdot x_{l_r},\ r \in [1, m], l_r \neq r_k && (E_r)\\
& && 2\sum\nolimits_{i=1}^{k-1} x_i + x_k \leq T && (B)
\end{aligned}
$$

Here, constraints $(C_0), (C_{j_0}), \dots (C_k)$ model the $(1 + 2\rho)$-competitiveness of the strategy, and constraint (B) models the budget constraint. See (Hipke et al. 1999) for the derivation of these constraints. Constraints $(E_1), \dots, (E_m)$ model the extendability property, by giving competitiveness constraints for targets placed just beyond the turn points at $x_{l_1}, \dots, x_{l_r}$. Indeed, once the search is completed, in order for it to be extendable, the searcher must be able to return just beyond the boundary of the cleared area, while remaining $(1 + 2\rho)$-competitive. For a point just beyond the searcher's final position this is trivially verified; for all other final turn points this incurs a competitiveness constraint, which has a similar form to the (C_j) constraint.

As is standard in star search problems, we can add some much-needed structure in the above formulation.

Theorem 1. *Any optimal solution $X^* = (x_i^*, r_i)$ to $L_m^{(k)}$ must be* monotone *and* cyclic*: (x_i^*) is increasing and $r_i = i \bmod m$ up to a permutation.*

This means that we can formulate the problem using a much simpler family of LPs which we denote by $P_m^{(k)}$, where constraints (M_i) model monotonicity.

$$
\begin{aligned}
&\max && \sum\nolimits_{i=0}^{m-1} x_{k-i} && (P_m^{(k)})\\
&\text{subj to} && \sum\nolimits_{i=1}^{m-1} x_i \leq \rho && (C_0)\\
& && \sum\nolimits_{i=1}^{j+m-1} x_i \leq \rho \cdot x_j, \qquad j \in [1, k-m] && (C_j)\\
& && \sum\nolimits_{i=1}^{k} x_i \leq \rho \cdot x_j,\ \ j \in [k-m+1, k-1] && (E_j)\\
& && x_i \leq x_{i+1}, \qquad i \in [1, k-1] && (M_i)\\
& && 2\sum\nolimits_{i=1}^{k-1} x_i + x_k \leq T && (B)
\end{aligned}
$$

Solving $P_m^{(k)}$

While proving cyclicality, we also prove that for any optimal solution to $L_m^{(k)}$, most of the constraints are tight, similarly to Lemma 1. Applying this result to $P_m^{(k)}$ gives the following.

Lemma 3. *In an optimal solution to the LP $P_m^{(k)}$, constraints (M_i) are not necessarily tight, at least one of the constraints (C_0) and (B) is tight, and all other constraints must be tight.*

Subtracting (C_i) from (C_{i+1}) and (C_{k-m}) from (E_{k-m+1}) gives a linear recurrence formula which any optimal solution X^* must satisfy:

$$x_{i+m}^* = \rho(x_{i+1}^* - x_i^*). \quad i \in [1, k-m]$$

The constraints (E_j) give us $m-1$ equations to help determine the solution: $\rho x_{k-m+1}^* = \dots = \rho x_{k-1}^* = S_k$. So X^*, viewed as a point in $\mathbb{R}^k$, is on a line $\Delta_m^{(k)} \subset \mathbb{R}^k$, defined as the set of all points which satisfy $(C_1), \dots, (E_{k-1})$ with equality. Lemma 3 shows that the solution to $P_m^{(k)}$ is either the point $X_0^{(k)} \in \Delta_k^m$ for which constraint (C_0) is tight, or the point $X_B^{(k)} \in \Delta_k^m$ for which constraint (B) is tight.

We can compute these two strategies efficiently for a fixed k, as we will demonstrate for $X_B^{(k)}$. We rewrite the conditions $X_B^{(k)} \in \Delta_k^m$ and "(B) is tight" as a matrix equation:

$$\mathcal{M}_{k,B}^m \times X = \begin{pmatrix} 0 & \cdots & 0 & T \end{pmatrix}^\top \qquad (3)$$

where $\mathcal{M}_{k,B}^m$ is the following $k \times k$ matrix:

$$\begin{pmatrix}
\rho & -\rho & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & \rho & -\rho & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \rho & -\rho & 0 \\
1 & 1 & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1-\rho & 1 \\
2 & 2 & 2 & 2 & \cdots & 2 & 2 & \cdots & 2 & 2 & 1
\end{pmatrix}$$

$\mathcal{M}_{k,B}^m$ has a very nice structure, and is very sparse, as all coefficients are concentrated in three diagonals (numbered 1, 2, and $m+1$) and the last two lines. This is good for us: we can solve (3) in time $O(k)$ using Gaussian elimination. $X_0^{(k)}$ can be computed similarly, using the matrix $\mathcal{M}_{k,0}^m$, which is identical to $\mathcal{M}_{k,B}^m$ except for the last line, which contains (C_0), and (3) becomes $\mathcal{M}_{k,0}^m \times X_0^{(k)} = (0 \cdots 0\, \rho)^\top$. When solving (3) we discarded the constraint (C_0), so we need to check whether $X_B^{(k)}$ is feasible for this constraint. Similarly, we need to check whether $X_0^{(k)}$ is feasible for (B).

Finding the Optimal Strategy

At this point, we have determined how to compute two families of strategies, the sets $\mathcal{X}_0 = \{X_0^{(k)}, k \in \mathbb{N}\}$ and $\mathcal{X}_B = \{X_B^{(k)}, k \in \mathbb{N}\}$, and we have shown that any optimal strategy belongs to one of these two families. Define k_0 the highest k for which $X_0^{(k)}$ is feasible, and k_B the lowest k for which $X_B^{(k)}$ is feasible. We conclude with our two main results.

Theorem 2. *$X_0^{(k)}$ is feasible if and only if $k \leq k_0$, and $X_B^{(k)}$ is feasible if and only if $k \geq k_B$. Moreover, $X_0^{(k_0)}$ is optimal for $\mathcal{X}_0$, and $X_B^{(k_B)}$ is optimal for $\mathcal{X}_B$.*

Proof sketch. We show first that any point (x_i) that is feasible for $P_m^{(k)}$ is positive: $\forall i, x_i \geq 0$. Denote $X_0^{(k)} = (x_i)$ and $X_0^{(k-1)} = (y_i)$. Using the convention $y_0 = 1$, the strategy $D = (x_i - y_{i-1})$ is feasible for P_m^k, therefore positive. This means that X has a higher objective value than Y, and also requires a larger budget: this shows that k_0 is well-defined and optimal. Because $X_0^{(k)}$ and $X_B^{(k)}$ are scaled versions of each other, we get $k_B = k_0$ or k_0+1. Additional calculations show that the objective values of $X_B^{(k)}$ are decreasing. □

Theorem 3. *The optimal strategy for the m-ray star can be computed in time $O(m \log(T) \log(m \log(T)))$.*

Proof sketch. The scaled geometric strategy with base $b = \frac{m}{m-1}$ is a feasible point for a certain $P_m^{(k_G)}$, with $k_G = O(\log_b(T)) = O(m \log(T))$. This means that $X_B^{(k_G)}$ is feasible, and so $k_B \leq k_G$ gives us an upper bound. We can use binary search to find k_B, solving (3) at each step at a cost of $O(k_G)$. We know that k_0 is either k_B or $k_B - 1$, so all that remains is to compare the two strategies, which gives us a total complexity of $O(m \log(T) \log(m \log(T)))$. □

Maximum Clearance in a Network

In this section we study the setting in which the environment is a *network*, represented by an undirected, edge-weighted graph $Q = (V, E)$, with a vertex O designated as the root. Every edge has a non-negative *length* which represents the distance of the vertices incident to the edge. The target can hide anywhere along an edge, which means that the search strategy must be a traversal of all edges in the graph. We can think of the network Q as being endowed with Lebesgue measure corresponding to the length. This allows as to define, for a given subset A of the network, its measure $l(A)$. Informally, $l(A)$ is the total length of all edges (partial or not) that belong in A. Given a strategy S and a target t, the cost $d(S,t)$ and the distance $d(t)$ are well defined, and so is the competitive ratio according to (1). We will denote by $Q[r]$ the subnetwork that consists of all points in Q within distance at most r from O.

The exact competitive ratio of searching in a network is not known, and there are only $O(1)$-approximations (Koutsoupias, Papadimitriou, and Yannakakis 1996; Angelopoulos and Lidbetter 2020) of the optimal competitive ratio. For this reason, as explained in the introduction, we interpret MAXCLEAR(R,T) as a maximum clearance strategy with budget T that is an *R-approximation* of the optimal competitive ratio. The known approximations use searching based on iterative deepening, e.g. strategy CPT(r), which in each round i, searches $Q[r^i]$ using a Chinese Postman Tour (CPT) (Edmonds and Johnson 1973) of $Q[r^i]$, for some suitably chosen value of r.

We could apply a similar heuristic to the problem of Maximum Clearance. However, searching using a CPT of $Q[r^i]$ is wasteful, since we repeatedly search parts of the network that have been explored in rounds $1 \ldots i-1$. Instead, we rely on heuristics for the *Rural Postman Problem* (Frederickson, Hecht, and Kim 1978). In this problem, given an edge-weighted network $Q = (V, E)$, and a subset $E_{\text{req}} \subseteq E$ of *required* edges, the objective is to find a minimum-cost traversal of all edges in E_{req}in Q; we call this tour RPT for brevity. Unlike the Chinese Postman Problem (CPP), finding an RPT is NP-hard. The best known approximation ratio is 1.5 (Frederickson, Hecht, and Kim 1978), but several heuristics have been proposed, e.g. (Corberán and Prins 2010), (Hertz, Laporte, and Hugo 1999).

We thus propose the following strategy, which we call RPT(r). For each round $i \geq 1$, let $R_{i-1} = Q[r^i] \setminus Q[r^{i-1}]$ denote the part of the network that the searcher has not yet explored in the beginning of round i (and needs to be explored). Compute both tours CPT($Q[r^i]$) and RPT($Q[r^i]$), the latter with required set of edges the edge set of R_{i-1} (using the 1.5-approximation algorithm), and choose the tour of minimum cost among them. This continues until the time

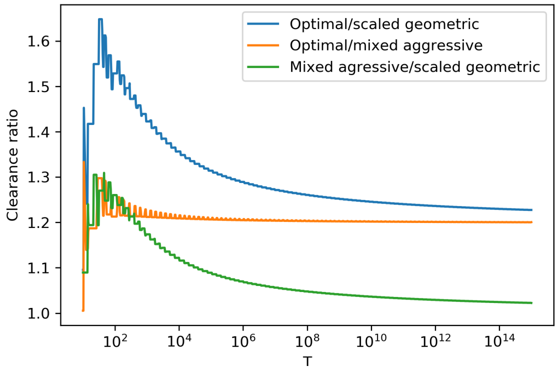

Figure 1: Clearance ratios for $m = 4$ and $R = R_4^*$, as function of T.

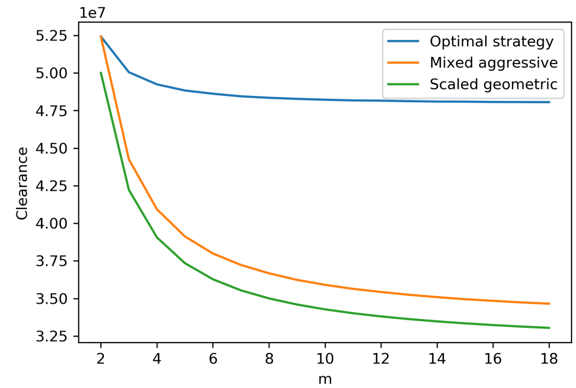

Figure 2: Clearance as function of m, for $T = 10^8$ and $R = R_m^*$.

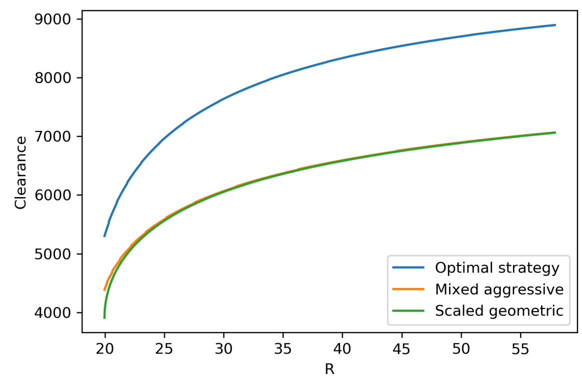

Figure 3: Clearance as function of R, for $m = 4$ and $T = 10^4$.

budget T is exhausted. It is very hard to argue from a theoretical standpoint that the use of RPT yields an improvement on the competitive ratio; nevertheless, the experimental evaluation shows that this is indeed beneficial to both competitiveness and clearance. Since RPT(r) is at least as good as a strategy that is purely based on CPTs, we can easily show the following, which is proven analogously to the randomized strategies of (Angelopoulos and Lidbetter 2020).

Proposition 1. *For every $r > 1$, RPT(r) is a $\frac{r^2}{r-1}$-approximation of the optimal competitive ratio. In particular, for $r = 2$, it is a 4-approximation.*

Note that RPT(r) is, by its statement, extendable, since it will always proceed to search beyond the boundary of round i in round $i+1$. Moreover, RPT(r) is applicable to unbounded networks as well, provided that for any D, the number of points in the network at distance D from O is bounded by a constant. This is necessary for the competitive ratio to be bounded (Angelopoulos and Lidbetter 2020).

Experimental Evaluation

m-ray Star

In this section we evaluate the performance of our optimal strategy against two other candidate strategies. The first candidate strategy is the scaled geometric strategy, with base ζ_2,which we consider as the baseline for this problem prior to this work. The second candidate strategy is the mixed aggressive strategy. Recall that we defined both strategies at the beginning of the star section, and that all these strategies are defined for the same competitive ratio R.

Figure 1 depicts the relative performance of the optimal strategy versus the performance of the other two strategies, for $m = 4$, and optimal competitive ratio $R = R_4^*$, for a range of budget values $T \in [10, 10^{15}]$. Once the budget T becomes meaningfully large (i.e, $T \geq 50$), the optimal strategy dominates the other two, outperforming both by more than 20%. In contrast, the mixed aggressive strategy offers little improvement over the scaled geometric strategy for every reasonably large value of T.

Figure 2 depicts the influence of the parameter m on the clearance achieved by the three strategies, for a relatively large value of $T = 10^8$. For each value of m in $[2, 18]$, we require that the strategies have optimal competitive ratio $R = R_m^*$. For the line ($m = 2$) we see that the mixed aggressive strategy is optimal. We observe that as m increases, each strategies' clearance decreases, however the optimal strategy is far less impacted. This means that as m increases, the relative performance advantage for the optimal strategy also increases, in comparison to the other two.

Figure 3 depicts the strategies' performance for $m = 4$, and $T = 10^4$, as a function of the competitive ratio $R \geq R_4^*$. In particular, we consider $R \in [R_4^*, 3R_4^*]$. We observe that as R increases, the mixed aggressive strategy is practically indistinguishable from the scaled geometric. The optimal strategy has a clear advantage over both strategies for all values of R in that range.

Networks

We tested the performance of RPT(r) against the performance of CPT(r). Recall that the former searches the network $Q[r^i]$ iteratively using the best among the two tours CPT($Q[r^i]$) and RPT($Q[r^i]$), whereas the latter uses only the tour CPT($Q[r^i]$). We found $r = 2$ to be the value that optimizes the competitive ratio in practice, as predicted also

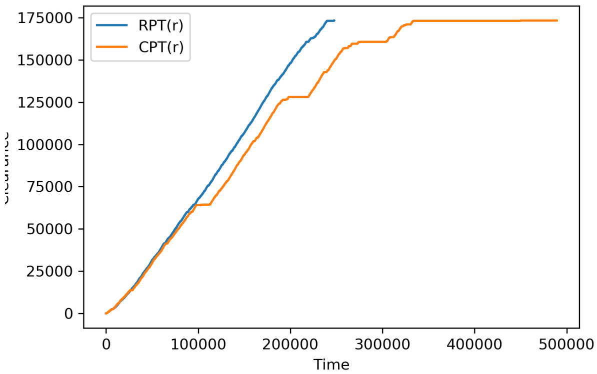

Figure 4: Comparison of the two strategies on the Berlin network (633 nodes, 1042 edges).

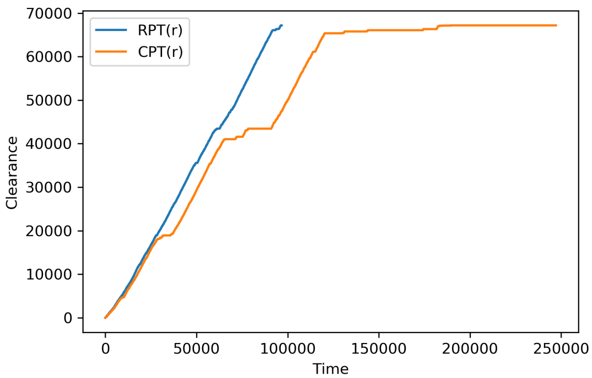

Figure 5: Comparison of the two strategies on the Chicago network (933 nodes, 1475 edges).

by Proposition 1, so we chose this value for our experiments.

We used networks obtained from the online library *Transportation Network Test Problems* (Bar-Gera 2002), after making them undirected. This is a set of benchmarks that is very frequently used in the assessment of transportation network algorithms (see e.g. (Jahn et al. 2005)). The size of the networks we chose was limited by the $O(n^3)$ time-complexity of CPT(r) and RPT(r) (n is the number of vertices). For RPT we used the algorithm due to (Frederickson, Hecht, and Kim 1978).

Figures 4 and 5 depict the clearance achieved by each heuristic, as function of the budget T, for a root chosen uniformly at random. The first network is a European city with no obvious grid structure, whereas the second is an American grid-like city. We observe that the clearance of CPT(r) exhibits plateaus, which we expect must occur early in each round, since CPT must then traverse previously cleared ground. We also note that these plateaus become rapidly larger as the number of rounds increases, as expected. In contrast, RPT(r) entirely avoids this problem, and performs significantly better, especially for large time budget.

Figure 6 depicts the ratio of the average clearance of RPT(r) over the average clearance of CPT(r) as a function of the time budget T, calculated over 10 random runs of each algorithm on the Berlin network (each run with a root chosen

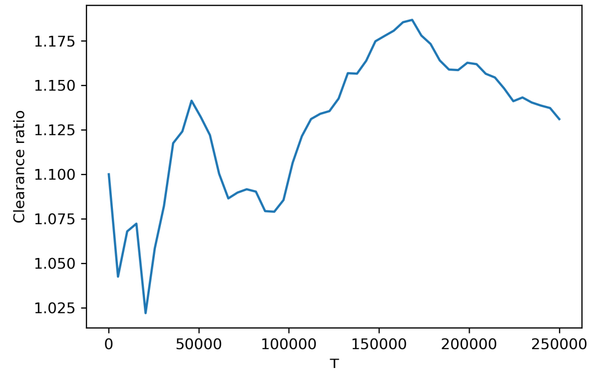

Figure 6: Clearance ratio of RPT(r) versus CPT(r), for 10 randomly chosen roots, for the Berlin network.

uniformly at random). We observe that RPT(r) consistently outperforms CPT(r), by at least 8% for most values of T, and up to 16% when T is comparable to the total length of all edges in the graph (173299). At $T = 250000$, in most runs, RPT(r) has cleared the entire network.

The average competitive ratios for these runs are 160 for CPT(r) and 132 for RPT(r), demonstrating a clear advantage.

Extensions and Conclusions

One can define a problem “dual” to Maximum Clearance, which we call Earliest Clearance. Here, we are given a bound L on the desired ground that we would like the searcher to clear, a required competitive ratio R, and the objective is to design an R-competitive strategy which minimizes the time to attain clearance L. The techniques we use for Maximum Clearance can also apply to this problem, in fact Earliest Clearance is a simpler variant; e.g., for star search, optimal strategies suffice to saturate all but one constraint, instead of all but two.

Maximum Clearance on a star has connections to the problem of scheduling *contract algorithms with end guarantees* (Angelopoulos and Jin 2019). More precisely, our LP formulation has certain similarities with the formulation used in that work (see the LP P_m, on page 5496 in (Angelopoulos and Jin 2019)), and both works use the same general approach: first, a technique to solve the LP of index k, and then a procedure for finding the optimal index k^*. However, there are certain significant differences. First, our formulations allow for any competitive ratio $\rho \geq \rho^*_m$, whereas (Angelopoulos and Jin 2019) only works for what is the equivalent of ρ^*_m. Related to this, the solution given in that work is very much tied to the optimal performance ratios, and the same holds for the optimality proof which is quite involved and does not extend in an obvious way to any ρ. The theoretical worst-case runtime of the algorithm in (Angelopoulos and Jin 2019) is $O(m^2 \log L)$, whereas the runtime of our algorithm has only an $O(m \log m)$ dependency on m, as guaranteed by Theorem 3.

References

Angelopoulos, S. 2015. Further connections between contract-scheduling and ray-searching problems. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 1516–1522.

Angelopoulos, S.; Dürr, C.; and Jin, S. 2019. Best-of-two-worlds analysis of online Search. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *LIPIcs*, 7:1–7:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Angelopoulos, S.; and Jin, S. 2019. Earliest completion scheduling of contract algorithms with end guarantees. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 5493–5499.

Angelopoulos, S.; and Lidbetter, T. 2020. Competitive search in a network. *European Journal of Operational Research* 286(2): 781–790.

Angelopoulos, S.; and Voss, M. 2020. Online Search with Maximum Clearance. *arXiv:2011.14144* .

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

Bar-Gera, H. 2002. Transportation Network Test Problems. URL https://github.com/bstabler/TransportationNetworks. Accessed on August 15, 2020.

Beck, A.; and Newman, D. 1970. Yet more on the linear search problem. *Israel Journal of Mathematics* 8: 419–429.

Bellman, R. 1963. An optimal search problem. *SIAM Review* 5: 274.

Bernstein, D.; Finkelstein, L.; and Zilberstein, S. 2003. Contract algorithms and robots on rays: unifying two scheduling problems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 1211–1217.

Bose, P.; Carufel, J. D.; and Durocher, S. 2015. Searching on a line: A complete characterization of the optimal solution. *Theoretical Computer Science* 569: 24–42.

Condon, A.; Deshpande, A.; Hellerstein, L.; and Wu, N. 2009. Algorithms for Distributional and Adversarial Pipelined Filter Ordering Problems. *ACM Transaction on Algorithms* 5(2): 24:1–24:34.

Corberán, A.; and Prins, C. 2010. Recent results on arc routing problems: An annotated bibliography. *Networks* 56(1): 50–69.

Easton, K.; and Burdick, J. 2005. A coverage algorithm for multi-robot boundary inspection. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 727–734. IEEE.

Edmonds, J.; and Johnson, E. L. 1973. Matching, Euler tours and the Chinese postman. *Mathematical programming* 5(1): 88–124.

Frederickson, G. N.; Hecht, M. S.; and Kim, C. E. 1978. Approximation algorithms for some routing problems. *SIAM Journal on Computing* 7(2): 178–193.

Gal, S. 1972. A general search game. *Israel Journal of Mathematics* 12: 32–45.

Gal, S. 1974. Minimax solutions for linear search problems. *SIAM Journal on Applied Mathematics* 27: 17–30.

Ghosh, S. K.; and Klein, R. 2010. Online algorithms for searching and exploration in the plane. *Computer Science Review* 4(4): 189–201.

Hertz, A.; Laporte, G.; and Hugo, P. N. 1999. Improvement procedures for the undirected rural postman problem. *INFORMS Journal on Computing* 11(1): 53–62.

Hipke, C.; Icking, C.; Klein, R.; and Langetepe, E. 1999. How to find a point in the line within a fixed distance. *Discrete Applied Mathematics* 93: 67–73.

Isler, V.; Kannan, S.; and Daniilidis, K. 2003. Local exploration: online algorithms and a probabilistic framework. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, 1913–1920. IEEE.

Jahn, O.; Möhring, R. H.; Schulz, A. S.; and Stier-Moses, N. E. 2005. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research* 53(4): 600–616.

Jaillet, P.; and Stafford, M. 1993. Online searching. *Operations Research* 49: 234–244.

Kao, M.-Y.; and Littman, M. 1997. Algorithms for informed cows. In *Proceedings of the AAAI 1997 Workshop on Online Search*.

Kao, M.-Y.; Reif, J.; and Tate, S. 1996. Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem. *Information and Computation* 131(1): 63–80.

Koutsoupias, E.; Papadimitriou, C.; and Yannakakis, M. 1996. Searching a fixed graph. In *Proc. of the 23rd Int. Colloq. on Automata, Languages and Programming (ICALP)*, 280–289.

Kupavskii, A.; and Welzl, E. 2018. Lower bounds for searching robots, some faulty. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018*, 447–453. ACM.

López-Ortiz, A.; and Schuierer, S. 2001. The ultimate strategy to search on m rays? *Theoretical Computer Science* 261(2): 267–295.

López-Ortiz, A.; and Schuierer, S. 2004. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. *Theoretical Computer Science* 310(1–3): 527–537.

Magid, E.; and Rivlin, E. 2004. CAUTIOUSBUG: A competitive algorithm for sensory-based robot navigation. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, 2757–2762. IEEE.

Sung, Y.; and Tokekar, P. 2019. A competitive algorithm for online multi-robot exploration of a translating plume. In *2019 International Conference on Robotics and Automation (ICRA)*, 3391–3397.

Taylor, C. J.; and Kriegman, D. J. 1998. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions on robotics and Automation* 14(3): 417–426.

Xu, L.; and Stentz, A. 2010. A fast traversal heuristic and optimal algorithm for effective environmental coverage. In *Robotics: Science and Systems VI, Universidad de Zaragoza, Zaragoza, Spain, June 27-30, 2010*. The MIT Press.