2010 American Control Conference
Marriott Waterfront, Baltimore, MD, USA
June 30-July 02, 2010

Whittle-Indexability of the Cow Path Problem

Tom Temple Emilio Frazzoli

Abstract— **In this paper we consider the well-studied Cow Path Problem (CPP), an on-line search problem that is typically treated with competitive analysis. This paper uses an alternative approach, posing the problem as a Markov Decision Problem (MDP). Our technical contribution is to prove that when posed as an MDP, a slightly relaxed version of the problem is *Whittle-indexable*, and we present the corresponding index heuristic. This result also provides an insight: theoretical properties that have been empirically vetted (such as the Whittle index) are a means to bridge the gap between theory and practice in on-line decision-making problems.**

I. Introduction

The Cow Path Problem (CPP), is an on-line search problem in which k short-sighted cows search for a reward (say, a patch of clover) on m paths which diverge from a single origin and never cross. The cows would like to find the reward while minimizing the time spent searching.

In on-line search problems such as the CPP, the most widely accepted theoretical framework is competitive analysis (see [1], [11] for surveys). In this framework, the goal is to determine a search strategy that minimizes the *competitive ratio*, which, informally, can be thought of as a worst-case constant-factor guarantee. Since it focuses on the worst-case, such a guarantee can lead to overly conservative strategies.

On the other hand, as was pointed out in [12], if we have (or assume) a prior distribution over the goal location, it is quite natural to pose search as a path planning problem. Typically, the path planning problem is treated as a Markov Decision Problem (MDP) in which agents sequentially choose actions in order to minimize expected cumulative cost. In continuous domains, solving the MDP to optimality is undecidable, in fact, it is PSPACE-hard even to approximate[13]. In particular, if the state space includes belief about hidden variables (*i.e.*, probabilities), then the problem is said to be a Partially Observable Markov Decision Problem (POMDP). Despite its difficulty, the POMDP has been extensively studied, widely used, and as a result, has had numerous practical successes. In light of these successes, this paper re-examines the CPP from the MDP framework.

The motivation for this paper comes from one such success: the Coastal Navigation Problem (CNP) presented in [14]. Those authors posed a path planning methodology that could intelligently exploit landmarks in order to reliably reach a goal location without solving a POMDP for the entire belief space. They did this by adding a single additional state variable that encapsulated the uncertainty, and then solving an MDP with the new joint state.

Consider, for example, the problem of returning to Ont's car, which is parked on a road, from the middle of the woods. Non-expert orienteers might be surprised that it is not optimal to aim straight for the goal, but rather to choose one side and deliberately overshoot. This is because it is more costly to have a ~ 0.5 probability of having to double back than it is to increase the length of the initial path. For each possible initial path, we must solve an CPP to determine the search policy once we reach the shore. Having done this, we can determine *by how much* the agent should initially overshoot.

If the CPP is a sub-problem of the CNP, and the MDP formulation of the CNP is practically effective, then it can be expected that such a formulation must also be effective on the CPP. This paper establishes the correctness of this intuition by formally proving that the CPP has a particular property, called *Whittle-indexability*, leading to the existence of a strong heuristic.

As a result we believe that the CPP provides an insightful example. It shows that properties like Whittle-indexability can be used to bridge the gap between the theoretical results in on-line search problems and the practical successes of on-line path planning.

A. Problem Statement

In the CPP, k agents are searching for a unique goal that lies on one of m rays diverging from a single origin (with $k < m$). We will call this set of locations the "region," $\mathcal{R} = \{1 \dots m\} \times \mathbb{R}_+$. The agents know their positions, and each has a sensor that can detect the goal only if the agent is at the location of the goal; otherwise it gives no information.[1] We would like to determine a search policy that finds the goal while minimizing the distance that the agents must travel.

In the usual formulation of the problem (see, *e.g.*, [7] and the references therein), the objective is to minimize the competitive ratio, Cr, of the search strategy. The competitive ratio of an on-line algorithm A is given by

$$\mathrm{Cr}_A = \sup_{\theta \in \Theta} \frac{c(A(\theta))}{c^*(\theta)}, \tag{I.1}$$

where Θ denotes the set of all problem realizations, $c(A(\theta))$ is the cost of the solution returned by A applied to realization θ, and $c^*(\theta)$ is the cost that could be found by an optimal offline algorithm with full knowledge of θ in advance.

For example, consider the case with one agent and two paths and assume that the goal location is known to be integer. From [1], the best-possible[2] schedule of turn-around

[1] While this sensor model might seem overly restrictive, solving problems using multiple agents and "minimalist" sensors is a topic of recent interest in robotics (see *e.g.*, [16], [15]).

[2] An on-line algorithm with the minimum achievable competitive ratio is called "best-possible" rather than "optimal" to avoid confusion with c^*, in Equation I.1.

points, $z(i)$, is $z(i) = (-2)^i$. If the goal is at distance n from the origin, the search takes $2(\sum_{i=0}^{\lfloor \lg n \rfloor} 2^i) + n$. The optimal offline algorithm would be able to travel straight there, so $c^* = n$. The competitive ratio is therefore

$$\sup_n \frac{2(\sum_{i=1}^{\lfloor \lg n \rfloor + 1} 2^i) + n}{n} = 9.$$

In contrast, in this paper we will try to minimize the *expected* distance travelled before reaching the goal. To do this we must have a probability distribution, $f : \mathcal{R} \to \mathbb{R}_{\geq 0}$, for the goal location. Given a prior distribution, it is straightforward to pose the CPP as a path planning problem that roboticists will find familiar.

One of the main arguments in favor of competitive analysis is that it does not have the requirement of a prior. However, from a Bayesian perspective, requiring a prior is not an unreasonable demand—there is nothing to prevent us from using an uninformative distribution. Furthermore, minimizing the expectation for the uninformative distribution is seldom equivalent to minimizing the competitive ratio. Regardless of how informative this prior distribution may be, this formulation is meaningfully distinct from that in Equation I.1.

We illustrate this difference by example. Consider the uniform distribution over the integers $-n, \ldots, n$. In the limit of $n \to \infty$ we have already discussed the best-possible competitive strategy, which turns around at $z(i) = -2^i$. This strategy searches for an expected distance of $33/8n$. The policy that simply searches each direction to the end, only searches for an expected distance of n—less than a quarter as far. As it happens, in the framework of expected distance, this policy is optimal and is returned by the algorithms we will subsequently develop.

Before we elaborate on this formulation, we make the following remark.

Remark 1.1: For algorithm A, with competitive ratio Cr_A, there is a probability measure over Θ such that $\mathop{\mathbb{E}}_{\Theta}[c(A)]$ is arbitrarily close to $\mathrm{Cr}_A \times c^*$. This is accomplished by assigning probability one to the instance (or an instance in the sequence) that maximizes Equation I.1. In other words, the competitive ratio provides a *tight* bound on performance in the worst-case over *probability distributions*.

B. Belief–state representation

The problem of path planning in the presence of uncertainty can be posed as a POMDP, which would require solving Bellman's Equation, shown in discounted, discrete form in Equation I.2.

$$V^*(s,b) = \max_{a \in A} \sum_{(s',b')} T(s',b'|s,b,a)(\beta V^*(s',b') + R(s,a,s')). \tag{I.2}$$

for actions A observed state s, belief state b, transition dynamics T, rewards R, and discount factor β.

The difficulty stems from the fact that the belief, b, lives in the space of all probability measures over $\mathcal{R}$. In the CPP however, the belief has a special structure that can be exploited in the solution of the problem. Due to the limited sensor model, either the goal is found or the posterior belief is the prior distribution with explored regions zeroed-out (and re-normalized). Hence for a given prior distribution, the belief can be represented by the limits of the explored regions, which we will refer to as the "frontier" z, which is simply a vector in $\mathbb{R}_+^m$.

This state representation is analogous to that used in [12] which showed that for one agent on two paths with finitely many discrete locations, the problem can be posed as a quadratic program and solved in time polynomial in the number of locations. However, if we were to extend this approach to m paths (while maintaining finitely many discrete locations), the time requirement would be exponential in m.

In the continuous, k-agent, m-path version examined here, the value function will not be the solution to a quadratic program, but rather a general non-linear partial differential equation, which is still quite difficult. At this point we could resort to approximation techniques but to do so will still be intractable; from [13] we know that any non-trivial approximation can still require time exponential in the size of our representation of the problem.

II. Bandit Problems

If we neglect to include the time taken moving through previously explored territory and pretend that success rates along different paths are independent, the CPP is equivalent to a Multi-Armed Bandit Problem (MABP). In the MABP, a single server must choose from between m processes exactly one on which to work. This problem is referred to as the "Multi-Armed Bandit" problem because of it's relevance to a gambler in front of a bloc of slot machines. Gittins famously showed in [8] that there exists an "index function," depending only on the state of a single process, which can be used to greedily solve the problem to optimality. Furthermore in the case of multiple agents, it is shown in [5] that it is optimal to pursue the k processes with the highest indices.

In this framework, the "state" is that of the processes, as opposed to that of the agents. In the CPP, "paths" naturally correspond to processes and we define the state of path i as the pair $(a_i, z_i) \in \{0,1\} \times \mathbb{R}_+$ where a_i indicates whether the path is being actively searched, and z denotes the "frontier" of the path. This lets us define the state of the entire problem as $(a, z) \in \{0,1\}^m \times \mathbb{R}_+^m$. For the treatment that follows, we will not explicitly track the locations of the agents and instead assume that there is an agent at frontier of path i if and only if $a_i = 1$.

The Multi-Armed Bandit problem has some relevant limitations. First, the state of unplayed "arms" (*i.e.*, processes) only changes when they are played. Secondly, there is no cost associated with switching between arms. If either of these features are present, the problem is called a Restless Bandit Problem (RBP) and Gittins' indexing policy is no longer optimal [4].

Nonetheless, Whittle in [18] used a a linear programming relaxation to derive an index *heuristic* that is available

if the problem has a property that is dubbed "(Whittle) indexability." There is a large and growing body of research that suggests that RBPs with this property are apparently *easier* than the general POMDP. Specifically, this property gives rise to a relatively simple policy which, in practice, often has a small optimality gap

The above discussion is an instance of a theoretically provable property of a problem being used to predict the performance of a heuristic. Such an approach could provide attractive ways for tackling problems in PSPACE-hard.

A. Whittle's index heuristic

For a single process let us define the *subsidy-γ problem* as the following MDP. The state space is the state of the process and there are two actions: the active and passive actions. For the active action, the transition function is that of the original process and for the passive action, the state does not evolve. The rewards are augmented by adding γ to the reward for the active action (from any state to any state).

Definition 2.1: Let $\Pi_i(\gamma)$ denote the set of states of the process for which the active action is optimal in the subsidy-γ problem. Process i is *indexable* if $\Pi_i(\gamma)$ increases monotonically from the empty set to the entire state space as subsidy, γ, increases from $-\infty$ to ∞. An RBP is said to be *indexable* if each process is indexable.

The *Whittle index*, γ_i^* of an indexable process i in state x is given by

$$\gamma_i^*(x) = \inf_{x \in \Pi_i(\gamma_i)} \gamma_i. \tag{II.1}$$

Whittle's *index policy* is to always pursue the process for which γ_i^* is minimum. While non-optimal in general, this heuristic has been extensively examined and has been shown to perform very well empirically, *i.e.*, within a few percent of optimal (see *e.g.*, [2], [6], [9]). Furthermore, the heuristic is asymptotically optimal as m goes to infinity [17]. As a result, there has been much recent effort into establishing the indexability of classes of problems.

One of the broadest such results is [10], which considers the case with discounted rewards and countably many states. Those authors showed that there exists an equivalent formulation that is indexable if the cost of switching between two processes can be decoupled into separate "tear-down" and "set-up" costs, depending only on the state of the processes being switched from and to, respectively. Although the CPP is continuous and undiscounted, the switching costs are of this form: They consist of the distance from the frontier being left to the origin (a tear-down cost) and the distance from the origin to the frontier to be explored (a set-up cost).

B. Relaxation

Results establishing indexability, while not requiring that unplayed arms do not evolve, universally require that the arms evolve *independently*. This property is also absent from the CPP. There is only one goal; finding it on a particular path tells a great deal about whether it will be found on a different path. In this paper we prove the Whittle-indexability of a slightly relaxed problem: we do not insist that there is exactly one goal. In particular, we assume that the probabilities of there being a goal at any two locations is independent. Specifically, for a given prior $f(x)$ on *the* goal, we convert the problem into one in which at any point there is probability $f(x)dx$ of finding *a* goal in a neighborhood of dx.

We conjecture that this relaxation does not unduly effect the quality of the heuristic based on the following rationale. The search policy is to select the paths with the highest indices. If the goal has not been found, then the dependence between different paths can be summarized in a normalization constant. Since this constant is shared between paths, the ordering of the paths will not be affected. Hence the policy will not be directly affected.

III. Indexability

We will establish indexability of the relaxed CPP by constructing a subsidy scheme for each branch that satisfies Definition 2.1. To do this, we assume that an agent pays a unit cost per unit distance moved, and we determine a system of subsidies that make it neutrally profitable for the agents to conduct the search.

We will define our subsidy, γ, such that an agent is paid only if it finds the goal. An equivalent formulation (which would be more semantically consistent with [18], [10]) would be one in which the reward is unitary and we penalize moving. We use this definition entirely for the intuitive appeal of agents bidding on a "search contract" in which we "pay" the agent to search for us.

Remark 3.1: Trivially, for $\gamma < 0$ the set of states for which it is profitable to move is surely empty since the agent receives no other rewards.

A. Switching Costs

If an agent switches from path i (with state $(1, z_i)$) to path j (with state $(0, z_j)$) the agent must move a distance $z_i + z_j$ before it can activate path j. This is problematic because we would like the subsidy to depend only on the state of the path being activated. We will address this problem by changing the costs analogously to [9], which has an intuitive interpretation in our formulation.

We change the costs as follows. For each active path the agent must maintain a "return counter" that always contains the cost of returning to the origin. This way when path i is abandoned, the cost of activating path j is only z_j. This reflects the true costs assuming that the agent will eventually return to the origin, which does not happen if the agent finds the goal. Therefore, we also must adjust the reward as follows. If an agent finds the goal, it gets the payment γ plus the value in the return counter.

Lemma 3.2: Assume that there is a non-zero probability of finding the goal on path i.[3] From any state, there exists a sufficiently large, finite subsidy $\gamma < \infty$, for which it is profitable to explore path i.

Proof: Let $f_i(x)$ denote the probability density of finding the goal at location x on path i. Let F_i denote the cumulative distribution of f_i.

[3]If the there is zero probability, we are free to ignore the path altogether.

The expected cost of exploring path i to distance d is upper-bounded by $2d$. The expected reward will be $(F(d) - F(z_i))\gamma$, which is non-zero for some $d > z_i$, by assumption. For such a d, we can set $\gamma > 2d/(F(d) - F(z_i))$ which guarantees that it is profitable to explore path i. ■

B. Subsidy Scheme

We are interested in finding the minimum subsidy, $\gamma_i^*(a_i, z_i)$, at which path i in state (a_i, z_i) becomes profitable[4]. Since we will only be considering a single path at a time, we will drop the subscripts.

For an agent exploring path with state (a, z) we can compute the expected cost of searching to frontier z'. For notational compactness let $F(x_1, x_2)$ denote $F(x_1) - F(x_2)$.

$$\begin{aligned}\mathbb{E}[c((a,z),z')] &= 2(1-a)z + \\ &\alpha\left(\int_z^{z'} \chi f(\chi)d\chi + (1-F(z',z))2(z'-z)\right)\end{aligned} \tag{III.1}$$

In the unrelaxed version of the problem, we know that there is exactly one goal. Therefore $\alpha = (1 - \sum_j F(z_j))^{-1}$ is a normalization term that depends on the the states of other paths. For the relaxed problem, $\alpha = 1$ and we will subsequently drop it.

This lets us define the minimum subsidy under which any search is immediately profitable

$$\gamma^*(a,z) = \inf_{z'>z} \frac{\mathbb{E}[c((a_i,z),z_i')]}{F(z_i', z_i)}. \tag{III.2}$$

We now define the states, $\Pi(\gamma)$, for which search is immediately profitable under γ.

$$\Pi(\gamma) \equiv \{(a,z) \text{ s.t. } \gamma \geq \gamma^*(a,z)\} \tag{III.3}$$

Remark 3.3: It is clear from Equation III.3 that $\gamma_0 \leq \gamma_1 \implies \Pi_i(\gamma_0) \subseteq \Pi_i(\gamma_1)$.

We proceed by deriving the optimal policy in the subsidy-γ problem. We can write Bellman's equation for the the expected reward of the optimal policy, V_γ^*, when started from state (a, z). Unlike Equation I.2, Equation III.4 uses the non-discounted form, which is well-defined since the optimal policy must find the goal with probability one.

$$\begin{aligned}V_\gamma^*(a,z) = \max\Bigg(0, \sup_{z'>z}\Big(&\gamma F(z',z) - \mathbb{E}[c((a,z),z')] + \\ &(1-F(z',z))V_\gamma^*(1,z')\Big)\Bigg)\end{aligned} \tag{III.4}$$

Lemma 3.4: Assume $V_\gamma^*(a,z) > 0$. For the frontier, z', that maximizes Equation III.4, $V_\gamma^*(1,z') = 0$, *i.e.*,

$$V_\gamma^*(a,z) = \max\left(0, \sup_{z'>z}\left(\gamma F(z',z) - \mathbb{E}[c((a,z),z')]\right)\right). \tag{III.5}$$

[4]We will use the word "profitable" to explicitly include the case in which the expected profit is zero.

Proof: Suppose to the contrary that $V_\gamma^*(1,z') > 0$ which is maximized by $z'' > z'$, *i.e.*,

$$\begin{aligned}V_\gamma^*(1,z') &= \gamma F(z'',z') - \int_{z'}^{z''} \chi f(\chi)d\chi + \\ &\quad (1-F(z'',z'))(2(z'-z'') + V_\gamma^*(1,z''))\end{aligned}$$

Expanding Equation III.4 and collecting terms we have

$$\begin{aligned}V_\gamma^*(a,z) &= \gamma F(z'',z) - 2(1-a)z - \int_z^{z''} \chi f(\chi)d\chi + \\ &\quad (1-F(z'',z))(2(z-z'') + V_\gamma^*(1,z'')) - \\ &\quad F(z',z)\left(\gamma F(z'',z') - \int_z^{z''} \chi f(\chi)d\chi\right) + \\ &\quad F(z'',z')(1-F(z,z'))2(z'-z'')\end{aligned}$$

From our assumption that $V_\gamma^*(a,z') > 0$, the second to last term must be negative. The last term is non-positive and zero only if there is no probability of finding the goal between z' and z''. We remove these terms and arrive at the inequality:

$$\begin{aligned}V_\gamma^*(a,z) &< 2(1-a)z + \gamma F(z'',z) - \int_z^{z''} \chi f(\chi)d\chi + \\ &\quad (1-F(z'',z))(2(z-z'') + V_\gamma^*(1,z'')).\end{aligned}$$

Since z', rather than z'', maximizes Equation III.4, we have the following contradiction

$$\begin{aligned}V_\gamma^*(a,z) &< \gamma F(z'',z) - \mathbb{E}[c((a,z),z'')] + V_\gamma^*(1,z'') \\ V_\gamma^*(a,z) &\geq \gamma F(z'',z) - \mathbb{E}[c((a,z),z'')] + V_\gamma^*(1,z'')\end{aligned}$$

■

Lemma 3.5: The optimal policy in the subsidy-γ problem is to search if and only if $(a,z) \in \Pi(\gamma)$.

Proof: [$\Longleftarrow$] We begin by proving that it is optimal not to search if $(a,z) \notin \Pi(\gamma)$

Substituting Equation III.2 into Equation III.5 and letting $\gamma = \gamma^* + \delta_\gamma$,

$$\begin{aligned}V_\gamma^*(a,z) = \max\Bigg(0, \sup_{z'>z}\Bigg(&\delta_\gamma F(z',z) - \mathbb{E}[c((a_i,z),z_i')] + \\ &\inf_{z''>z} \frac{\mathbb{E}[c((a,z),z'')]}{F(z'',z)} F(z',z)\Bigg)\Bigg)\end{aligned} \tag{III.6}$$

Noting

$$\inf_{z''>z} \frac{\mathbb{E}[c((a,z),z'')]}{F(z'',z)} \leq \frac{\mathbb{E}[c((a,z),z')]}{F(z',z)}, \tag{III.7}$$

we can conclude that

$$V_\gamma^*(a,z) \leq \max\left(0, \sup_{z'>z}(\delta_\gamma F(z',z))\right). \tag{III.8}$$

Since $(a,z) \notin \Pi(\gamma)$ implies $\delta_\gamma < 0$, we can conclude that $V_\gamma^*(a,z) = 0$. Therefore the passive action is optimal. ■

Proof: [$\Longrightarrow$] We now prove that for $(a,z) \in \Pi(\gamma)$ it is optimal to search. We divide this into two cases.

Case 5.1: Let $(a,z) \in \Pi(\gamma)$ and assume that the infimum in Equation III.2 is achieved by some $z'' > z$

Proof: [Proof of Case 5.1] By assumption,

$$\gamma^*(a,z) = \frac{\mathbb{E}[c((a,z),z'')]}{F(z'',z)}.$$

The supremum in Equation III.6 is over z' rather than z''; hence

$$V^*_\gamma(a,z) \geq \max\left(0, (\delta_\gamma F(z'',z))\right).$$

Since $(a,z) \in \Pi(\gamma)$ implies $\delta_\gamma \geq 0$, we can conclude that the active action is optimal. ■

Case 5.2: Let $(a,z) \in \Pi(\gamma)$ and assume that the infimum in Equation III.2 is achieved by the limit of some sequence $z^i = z^\infty$. We divide this into three sub-cases.

Case 2.1: $z^\infty = \infty$. In this limit, the cost of searching to z^i is unbounded, therefore γ^* must be unbounded and δ_γ cannot be positive. This case cannot occur.

Case 2.2: $z < z^\infty < \infty$

Proof: The expected cost of searching from z to $z' > z$ is continuous in z'. Hence the cost of searching the open interval $[z, z^\infty)$ is equal to that of searching its closure. At the same time, the expected reward is non-decreasing in z'. Therefore this case entails Case 5.1. ■

Case 2.3: $z^\infty = z$

Proof: We rewrite Equation III.7, with $\lim_{i\to\infty} z^i = \lim_{dz\to 0^+} z + dz$

$$\gamma^*(a,z) = \lim_{dz\to 0^+} \frac{\mathbb{E}[c((a,z),z+dz)]}{f(z)dz}$$

Since $\sup_{x>0} g(x) \geq \lim_{x\to 0} g(x)$ we can rewrite Equation III.6 as

$$\begin{aligned} V^*_\gamma(a,z) \geq \quad & \max\left(0, \lim_{dz\to 0^+}\left(\delta_\gamma f(z)dz + \gamma^*(a,z)f(z)dz - \right.\right. \\ & \left.\left. \frac{\mathbb{E}[c((a,z),z+dz)]}{f(z)dz}\right)\right) \\ \geq \quad & \max\Big(0, \lim_{dz\to 0^+} \delta_\gamma f(z)dz\Big). \end{aligned}$$

Since f and dz are non-negative, $\delta_\gamma \geq 0$ implies that it is optimal to search. ■

Since these cases are exhaustive, this concludes the proof of Lemma 3.5 ■

Theorem 3.6: The relaxed CPP is indexable.

Proof: Lemma 3.5 proves that $\Pi(\gamma)$ is exactly the set of states for which the active action is optimal in the subsidy-γ problem. From Remark 3.1, Lemma 3.2, and Remark 3.3 we have established that $\Pi(\gamma)$ increases monotonically from the empty set to the entire state space as γ goes from $-\infty$ to ∞, satisfying Definition 2.1. ■

Corollary 3.7: The Whittle index (defined in Equation II.1) is exactly $\gamma^*(a,z)$, given by Equation III.2. This follows directly from the definition of $\Pi(\gamma)$ given by Equation III.3.

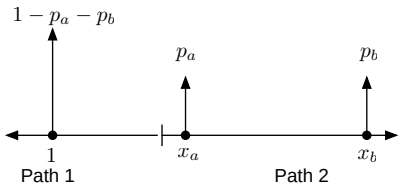

Fig. 1. Prior belief for three impulses on two paths.

IV. Examples

A. Toy example

From the nature of PSPACE-hard problems, any numerical evaluation is going to be essentially anecdotal. This is precisely why we have embraced the literature on Whittle indexability. Nonetheless, we think the following example provides an intuition while examining an interesting portion of the problem space. This example will show the heuristic's non-optimality, but will also demonstrate that it is a good approximation.

Suppose that there are two paths and the prior belief consists of three impulses, as shown in Figure 1. Note that one impulse is placed at unit distance without loss of generality.

First, for simplicity, we let $p_a = p_b = 0.25$ and let $x_b = 2 - x_a$. Let $\pi_{12}, \pi_{21}, \pi_{212}$ denote the three possible search policies that search according to the order of their subscripts. We compute the expected costs[5]

$$\begin{aligned} c_{12} &= (1-p_a-p_b) + p_a(2+x_a) + p_b(2+x_b) = 2 \\ c_{21} &= p_a(x_a) + p_b(x_b) + (1-p_a-p_b)(2x_b+1) \\ &= 3 - x_a \\ c_{212} &= p_a(x_a) + (1-p_a-p_b)(2x_a+1) + \\ &\quad p_b(2x_a+2+x_b) \\ &= 1.5 + 1.5x_a. \end{aligned}$$

We compute the minimum subsidies

$$\begin{aligned} \gamma^*_1 &= \frac{(1-p_a-p_b)+(p_a+p_b)2}{1-p_a-p_b} = 3 \\ \gamma^*_2 &= \min\Big(\frac{p_a x_a + (1-p_a)2x_a}{p_a}, \\ &\qquad \frac{p_a x_a + p_b x_b + (1-p_a-p_b)2x_b}{p_a+p_b}\Big) \\ &= \min\left(7x_a, 5.5 - 2x_a\right). \end{aligned} \tag{IV.1}$$

We will later refer to the left and right terms of Equation IV.1 as γ_{2a} and γ_{2b}, respectively.

If we explore path 2 to x_a, γ^*_2 will change to

$$\gamma_{2ab} = \frac{p_b(x_b - x_a) + (1-p_b)2(x_b-x_a)}{p_b} = 14(1-x_a).$$

Comparing the costs, we can see that policy π_{212} is optimal for $0 \leq x_a \leq 1/3$ and policy π_{12} is optimal for $1/3 \leq x_a \leq 1$. Comparing the subsidies we see that the

[5]Note that the expected costs are those from the original, unrelaxed problem.

index policy is optimal except for $1/3 < x_a < 3/7$. In this range the worst case is $15/14$ of optimal, a sub-optimality of about 7%.

We now remove the restrictions on p_a, p_b and x_b and consider all possible priors over three discrete locations on two paths. We will optimize over the locations and priors in order to find the greatest sub-optimality.

From this we construct six optimization problems: one for each possible ratio of the costs. Each problem is subject to constraints on γ_1, γ_{2a}, γ_{2b}, and γ_{2ab} such that the numerator is the strategy chosen by the index policy. By way of example, Problem IV.2 assumes that policy π_{21} is optimal, but the index policy instead pursues π_{12}.

$$\max_{x_a,x_b,p_a,p_b \geq 0} \frac{c_{12}}{c_{21}} \quad \text{s.t. } \gamma_1 < \gamma_{2a}, \quad \gamma_1 < \gamma_{2b}, \quad x_a \leq x_b, \quad p_a + p_b \leq 1 \qquad \text{(IV.2)}$$

To solve these problems we used Matlab's FMINCON search using the "active-set" algorithm, with 400 random restarts. These were divided into four sets of one-hundred cases, where x_a, x_b were chosen from an exponential distribution with scale parameters of 10, 100, 1000, or 10000. The largest sub-optimality was 38% which occurs at $x_a = 7.2$, $x_b = 23.2$, $p_a = 0.56$, and $p_b = 0.33$ and when the index policy chooses strategy π_{212} although π_{12} is optimal. Of course, this does not prove a lower-bound. However, it does show that if it *is* possible to achieve greater than 38% sub-optimality, such cases must be rare if we are unable to find them using this search technique.

We compare this to the worst-case sub-optimality (over x_a, x_b, p_a, p_b) of the best-possible competitive algorithm. It's easy to show that for $x_a = x_b = 1$, the best-possible competitive ratio is three. From Remark 1.1, p_a, p_b can be chosen such that this algorithm will be a factor of three from optimal. Therefore the worst-case is at least this great.

Compared to a factor of three, we can interpret a 38% sub-optimality as strong performance. Notably however, it a substantially greater level of sub-optimality than is commonly reported in numerical studies on indexable problems[2], [6], [9]. This suggests that the relaxation described in Section II-B is playing a non-negligible role. Future work will attempt to characterize this role.

B. Larger example

In this section we compare the best-possible competitive policy to the index policy for distributions with continuous support. It is clear that the index policy stands to benefit from knowledge of the prior. To provide the most fair comparison, therefore, we will use the least informative distributions: uniform and exponential.

Since these distributions are continuous at zero we must modify our definition of the competitive ratio. If we discretize the region with fidelity ε, the best-possible deterministic policy (from [3]) turns at

$$\left(\frac{m}{m-1}\right)^i \varepsilon. \qquad \text{(IV.3)}$$

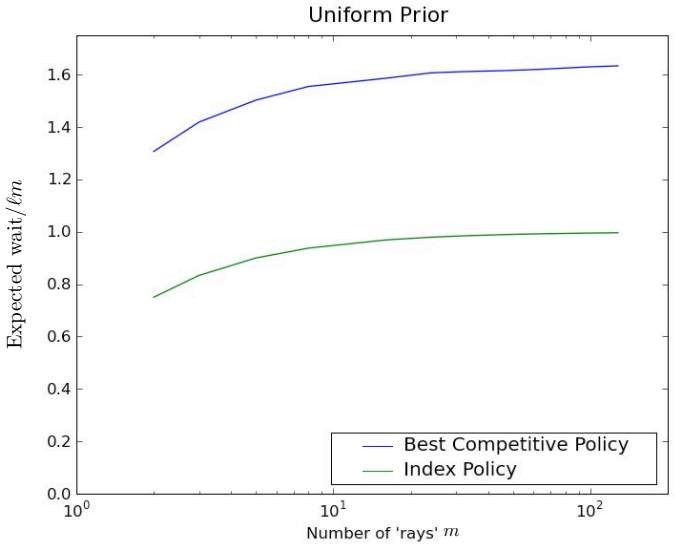

Fig. 2. Expected wait times for the uniform prior

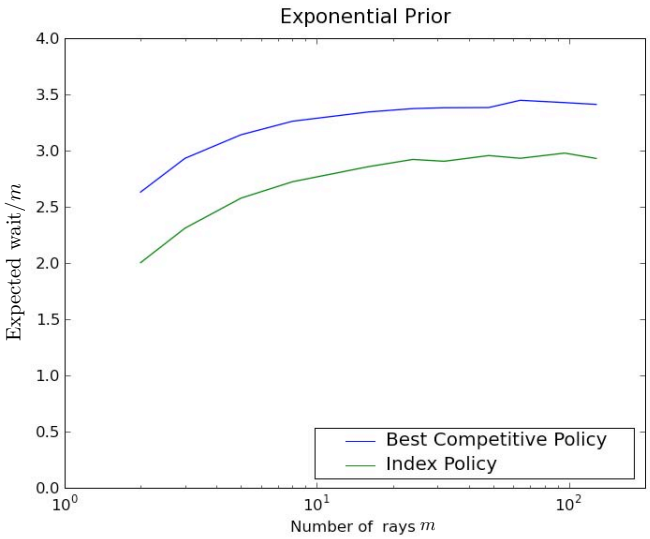

Fig. 3. Expected wait times for the exponential prior

For the analysis here we consider the limit of $\varepsilon \to 0$, in which the agent travels a distance of $2m|x|$ before turning at x.

The uniform prior with infinite support is improper. Instead, we consider uniform probability over finite support ℓ while taking the limit

$$\lim_{\ell \to \infty} \frac{\mathbb{E}_{x \sim U(0,\ell)^m}[w(x)]}{\ell}$$

For the index policy, this evaluates to $(m - 1/2)$. For the best-competitive policy this limit evaluates to $47/18$ at $m = 2$. For higher values of m we resort to numerical simulation, the results of which are presented in Figure 2.

Figure 3, shows simulation results for the exponential prior. In this case we immediately resort to simulation, using importance sampling to improve the estimate.

In both cases, the index policy significantly outperforms the competitive policy. This is notable because we selected priors that we expected to most favor the competitive policy. These results illustrate the fact that, even with uninformative priors, minimizing the competitive ratio does not minimize the expected wait time.

V. Conclusion

This paper formulates the Cow Path Problem (CPP) within the MDP framework and describes an algorithm for finding policies that minimize the expected search time. While this algorithm is non-optimal we have proved that the relaxed CPP is Whittle-indexable, which suggests that the optimality gap is small. This is in contrast to existing work on the CPP which predominantly seeks policies with minimal competitive ratios. While these ratios are guarantees, the resulting algorithms are very conservative. In particular we show that even with uninformative priors, the policies that minimize the competitive ratio provide poorer average service than the index policy developed here.

This paper leverages a growing body of work that identifies sub-classes of the POMDP that, empirically, are not as hard as the general case. In addition, we contribute to this body by proving the Whittle-indexability of a

- transition system with terminal states and non-discounted rewards, and with
- set-up and tear-down costs in a continuous domain.

Although it is a difficult path planning problem, the CPP is not a particularly *dynamic* problem. Future work will consider a larger class of stochastic dynamic programming problems that arise in on-line decision-making. In particular, we hope to demonstrate the applicability of Whittle's index and other well-studied heuristics to more general Dynamic Vehicle Routing Problems.

Acknowledgments

This research was done with support from the Michigan/AFRL Collaborative Center on Control Sciences, AFOSR grant no. FA 8650-07-2-3744. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the supporting organizations.

References

[1] S. Albers. Online algorithms: a survey. *Mathematical Programming*, 97(1):3–26, 2003.

[2] P. S. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O'Keeffe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57(1):21–39, 2003.

[3] R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, October 1993.

[4] J. S. Banks and R. K. Sundaram. Switching costs and the gittins index. *Econometrica*, 62(3):687–94, 1994.

[5] D. Bertsimas and J. Niño-Mora. Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems. *Mathematics of Operations Research*, pages 257–306, 1996.

[6] D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, pages 80–90, 2000.

[7] Rudolf Fleischer, Tom Kamphans, Rolf Klein, Elmar Langetepe, and Gerhard Trippen. Competitive online approximation of the optimal search ratio. *The 12th Annual European Symposium on Algorithms*, pages 335—346, 2004.

[8] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, 41(2):148–177, 1979.

[9] K. D. Glazebrook, H. M. Mitchell, and P. S. Ansell. Index policies for the maintenance of a collection of machines by a set of repairmen. *European Journal of Operational Research*, 165(1):267–284, August 2005.

[10] K. D. Glazebrook, D. Ruiz-Hernandez, and C. Kirkbride. Some indexable families of restless bandit problems. *Advances in Applied Probability*, 38(3):643, 2006.

[11] P. Jaillet and M. R. Wagner. *The Vehicle Routing Problem: Latest Advances and New Challenges*, chapter Online Vehicle Routing Problems: A Survey. Operations Research Computer Science Interfaces. Springer, 2008.

[12] M. Y. Kao and M. L. Littman. Algorithms for informed cows. In *AAAI-97 Workshop on On-Line Search*, 1997.

[13] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, pages 441–450, 1987.

[14] N. Roy and S. Thrun. Coastal navigation with mobile robots. *Advances in Neural Processing Systems*, 12:1043–1049, 1999.

[15] S. Suri, E. Vicari, and P. Widmayer. Simple robots with minimal sensing: From local visibility to global geometry. *The International Journal of Robotics Research*, 27(9):1055, 2008.

[16] B. Tovar, L. Freda, and S. M. LaValle. Using a robot to learn geometric information from permutations of landmarks. *Contemporary Mathematics*, 438:33–45, July 2007.

[17] R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of applied probability*, 27(3):637–648, 1990.

[18] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.