DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD

Incentive Design for Adaptive Agents

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters.

Citation	Chen, Yiling, Jerry Kung, David C. Parkes, Ariel D. Procaccia, and Haoqi Zhang. Forthcoming. Incentive Design for Adaptive Agents. In AAMAS '11: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, ed. Pnar Yolum, Kagan Tumer, Peter Stone and Liz Sonenberg, 627-634. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
Published Version	http://www.ifaamas.org/Proceedings/aamas2011/papers/B5_R62.pdf
Accessed	February 19, 2015 8:39:37 AM EST
Citable Link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:4912779
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP

(Article begins on next page)

Incentive Design for Adaptive Agents

Yiling Chen
School of Engineering and
Applied Sciences
Harvard University
Cambridge, MA 02138 USA
yiling@eecs.harvard.edu

Jerry Kung
School of Engineering and
Applied Sciences
Harvard University
Cambridge, MA 02138 USA
jkung@fas.harvard.edu

David C. Parkes
School of Engineering and
Applied Sciences
Harvard University
Cambridge, MA 02138 USA
parkes@eecs.harvard.edu

Ariel D. Procaccia
School of Engineering and
Applied Sciences
Harvard University
Cambridge, MA 02138 USA
arielpro@seas.harvard.edu

Haoqi Zhang
School of Engineering and
Applied Sciences
Harvard University
Cambridge, MA 02138 USA
hq@eecs.harvard.edu

ABSTRACT

We consider a setting in which a principal seeks to induce an adaptive agent to select a target action by providing incentives on one or more actions. The agent maintains a belief about the value for each action—which may update based on experience—and selects at each time step the action with the maximal sum of value and associated incentive. The principal observes the agent's selection, but has no information about the agent's current beliefs or belief update process. For inducing the target action as soon as possible, or as often as possible over a fixed time period, it is optimal for a principal with a per-period budget to assign the budget to the target action and wait for the agent to want to make that choice. But with an across-period budget, no algorithm can provide good performance on all instances without knowledge of the agent's update process, except in the particular case in which the goal is to induce the agent to select the target action once. We demonstrate ways to overcome this strong negative result with knowledge about the agent's beliefs, by providing a tractable algorithm for solving the offline problem when the principal has perfect knowledge, and an analytical solution for an instance of the problem in which partial knowledge is available.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics, Theory

Keywords

Coordination, economically-motivated agents, multiagent systems, principal-agent problem

Cite as: Incentive Design for Adaptive Agents, Yiling Chen, Jerry Kung, David C. Parkes, Ariel D. Procaccia, and Haoqi Zhang, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX–XXX.

Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1. INTRODUCTION

Many situations arise in which a principal wishes to affect the decisions of an agent as he learns to make decisions. For example, a teacher wishes for a student to check answers. A coach wishes for an athlete to adopt particular techniques. A marketer wants a consumer to purchase a particular brand of a product. In these examples, an agent's belief about his valuation for available actions may change with experience through learning or other forms of belief updates. The student may initially check answers but notice that this is time consuming and stop before he becomes good at it. The athlete may adopt and improve a nevertheless imperfect technique and keep with it. The consumer may purchase another brand and develop a loyalty to that brand.

We consider problems in which the *principal* can provide incentives to lead the agent to select a desired action. The teacher can provide gold stars for students who check their answers. The coach can spend effort on teaching a preferred technique. The marketer can advertise or offer discounts on a product. In some cases the provided incentives may not only change the agent's current selection, but also the agent's future selections because he learns that a particular action has high intrinsic value.

We conceptualize this problem as *incentive design for adaptive agents*. An agent's decision problem is assumed to be a multi-armed bandit problem [9, 6]. The agent selects a single action at each time step, and only its belief on the value of that action may change. In addition to modeling learning agents, this models sequential decision problems in which an agent's value for an action adapts over time; e.g., a new toy loses appeal over time or becomes damaged, or a task is completed and an action no longer has value.[1] The principal can provide incentives to influence the agent's behavior, with the goal of inducing a desired action once or multiple times. We insist that the incentives do not affect an agent's (intrinsic) belief on the value of each action, conditional on actions taken.

In our main formulation, the principal has no information about the agent's beliefs on value. But we also consider

[1] We will sometimes use 'learning' to describe the behavior of the agent in the sequel, but intend for such descriptions to also apply to agents with more general adaptive processes.

a variant where the principal is informed. Without knowledge, the problem is to use a limited budget to induce a desired behavior even though incentives can have different consequences when provided at different times. The use of incentives is also somewhat limiting, in that we cannot force the agent to select a particular action.

Our results. We consider two settings, one in which the principal has a fixed budget at each time step and another where the principal has a fixed budget across time steps. In the case of a fixed budget at each time step, we show that the quickest way for a principal to induce a target action once is to assign the budget to this action and wait for the agent to want to select the action. This is optimal for any update process and even with complete knowledge of the update process. Thus, it is optimal for Bayesian learners, as well as heuristic learners, that fit within our general framework. We think this is an interesting finding: the agent's belief update process is left unchanged until the point at which the agent can be incentivized to select the target action. This incentive scheme is also optimal for inducing the goal action to be selected as many times as possible within a fixed number of time periods.

In the case where the principal has a fixed budget across time, the problem is further complicated because the principal needs to decide when to spend the budget. For inducing the target once, assigning the entire budget to the target action remains optimal even with knowledge of the update process. Since no money is spent when the target is not selected, this policy remains feasible for a fixed budget and is therefore optimal for this more constrained problem. But for inducing the goal multiple times, we show that without knowledge of future values, no deterministic or even randomized algorithm provides a bounded competitive ratio for approximating the optimal offline solution, that is, the one obtained when given knowledge of the entire belief sequence. We show that a tractable algorithm exists for finding optimal incentives in the offline problem, and demonstrate on a particular instance of the problem how partial knowledge about the update process and beliefs over values can be used for finding effective incentives.

Related work. In terms of designing incentives to influence an agent's behavior when the agent's preferences are unknown, this work is related to work by Zhang et al. [12, 13, 11] on *environment design* and *policy teaching. Environment design* considers the problem of perturbing agent decision problems in order to influence their behavior. *Policy teaching* considers the particular problem of trying to influence the policy of an agent following a Markov Decision Process by assigning rewards to states. In these papers the agent is assumed to have a particular way of making decisions and persistent preferences. This paper can be seen as part of a larger agenda of *online environment design*, where a principal aims to make limited changes to an environment so as to influence the decision of agents while their valuations are still changing, possibly due to learning.

We are not aware of any work on bandits problems that considers a principal who through incentives seeks to induce an adaptive agent to learn to select an action that is desired by the principal.[2] The most closely related work is by Stone and Kraus [10] on *ad hoc* teams. In an *ad hoc* team, there is a learner with values for actions that update based on the empirical mean of observed values, and a teacher who intervenes by taking actions, which lead the agent to make another observation and update its beliefs. The goal is to maximize the combined performance of the teacher and the learner. The main finding is that it is never optimal to teach the worst arm, notably because teaching this is costly and the agent learns that this is the worst arm on its own at no additional loss. On surface level, this seems similar to our positive result on providing incentives on the target action: our agent must learn on its own that the other actions are not as good. However, our setting is quite different in that we cannot directly demonstrate a particular action to the agent but must intervene through incentives. Moreover, the principal's goal need not be aligned with that of the agent, and is ignorant of the agent's values or update process, which can be arbitrary.

Brafman and Tennenholtz [4] consider a teaching setting where a teacher can perform actions within a game to influence the behavior of a learner. However, in this setting there are no incentives and for the most part there is no cost to teaching.

Our problem is also somewhat similiar to the problem of reward shaping within reinforcement learning, where the goal is to adjust an agent's reward feedback in order to improve its performance in a complex environment [8, 7]. However, the assumptions we make are quite different. For example, the agent is not programmable, its values are not observed, and the shaping rewards are costly.

2. THE BASIC MODEL

We consider an agent with a set of actions $K = \{1, \ldots, n\}$. Let $K_{-i} = K \setminus \{i\}$. We use discrete time $t \in \{1, 2, \ldots\}$, and assume that the agent's belief about his value for an action at time t is dependent only on its state $x_i(t)$, which represents the agent's experience with action i prior to time t. Let $v_i(x_i(t))$ denote the agent's belief of the value of action i at time t if selected. At each time step t, the agent selects a particular action i, whose state transitions from $x_i(t)$ to $x_i(t+1)$, independently of time and the states of other actions. This transition can be stochastic, and for example can depend on the sequence of realized rewards from experiences with a particular action. The states of all other actions stay fixed, i.e., $x_j(t+1) = x_j(t), \forall j \neq i$. Throughout the paper, we find it notationally convenient to refer to the state of action i after it has been selected k times as x_i^k, and the agent's belief about its value as $v_i(x_i^k)$.

The agent's current belief can be an arbitrary function of the state, and thus can represent a range of adaptive agent behaviors. This includes, for example, an agent that selects an action according to the empirical average of rewards drawn so far, perhaps coupled with variance weighting to encourage exploration. To illustrate, let $r_1^i, \ldots, r_k^i$ denote the realized rewards received from each of the first k selections of action i. To encode an agent whose belief is the empirical average of rewards, let $v_i(x_i^k) = (\sum_{j=1}^k r_j^i)/k$ for $k \geq 1$.

[2] Cavallo et al. [5], Bergemann and Välimäki [2] and Babaioff et al. [1] study a distinct model of incentives in multi-armed bandit problems, from the mechanism design perspective. Each arm is associated with a different agent, and agents have private information about the rewards behind the arms. The goal is to design truthful mechanisms that elicit this information, and enable the center to utilize policies for selecting which arm to pull next to (approximately) maximize social welfare.

To encode the belief of an agent making explore and exploit tradeoffs, we can for example let

$$v_i(x_i^k) = d(x_i^k) + (\sum_{j=1}^{k} r_j^i)/k,$$

where $d(x_i^k)$ is the expected variance in rewards received from selecting action i and is decreasing in k. Similarly, Bayesian learning can also be directly modeled.

We consider a principal who wishes for the agent to select a target action g. The principal can provide incentives $\Delta(t) = (\Delta_1(t), \ldots, \Delta_n(t))$ at each time t, where $\Delta(t)$ can in general depend on any knowledge available to the principal, such as the incentives provided and actions selected prior to time t. The agent observes $\Delta(t)$ prior to selecting his action at time t, and the selected actions are observed by the principal. We assume that incentives are not incorporated into the agent's state, that is, the evolution of an agent's beliefs are independent of the incentives we offer, conditioned on the action the agent selects. We let $\Delta = (\Delta(1), \Delta(2), \ldots, \Delta(t), \ldots)$ denote a sequence of incentive decisions, which are induced by an *incentive policy*. Unless otherwise specified, we assume the principal has no knowledge of the agent's update process, and does not observe the realized rewards from the agent's selections.

In each time period, the agent selects the action with the maximal combined value using the following agent function:[3]

$$f(x(t), \Delta(t)) = \operatorname*{argmax}_{i \in K} [v_i(x_i(t)) + \Delta_i(t)]. \qquad (1)$$

The agent is myopic with regard to the intervention of the principal, in that the agent selects the action with the highest combined value without considering the effect of its action on future incentive provisions. Equivalently, the agent adopts a belief that the external incentive is exogeneous and invariant to its own policy, and thus something that does not need to be modeled. While myopic with respect to future incentives, the agent's choice can still reflect explore vs. exploit tradeoffs in its intrinsic value as explained above. However, by assuming that incentives are not incorporated into agent's state, we preclude models of learning in which an agent 'internalizes' the incentives over time.

The online model. Our main analysis is carried out in an online model of computation (see, e.g., [3]); for our purposes an informal description suffices. An instance of our problem specifies a sequence of belief value updates $v_i(x_i^0), v_i(x_i^1), \ldots$, for each action $i \in K$ and, optionally, a number of periods R. We assume that the principal has no knowledge of these values, and for the most part achieve incentive policies that could not be improved even with full knowledge. Our goal is to design algorithms with the same performance as the optimal offline algorithm with full knowledge of the input. As is usual, we will seek to compete in this sense with the offline algorithm even if the next value of each action is determined after each action of the algorithm in a way that is adversarial and dependent on the history. The performance is measured with respect to one of several objective criteria that we define in the sequel.

3. PER-PERIOD BUDGET

We consider first a principal that has a fixed budget at each time step. For example, consider a teacher with a limit of giving two gold stars per period, a coach with a fixed amount of time to demonstrate a preferred technique each period, or a marketer with a cap on the amount of discount that can be provided to a consumer across a set of products. For a per-period budget $B > 0$, we define the budget constraint on Δ as $\Delta_i(t) \leq B$ for all t and $i \in K$, and require further that incentives are non-negative, such that $\Delta_i(t) \geq 0$ for all actions i and times t. Note that the budget constraint formulation assumes that incentives are provided to the agent if and only if the agent selects the action with incentives applied to that action. This captures scenarios where incentives represent contracts (e.g., if you buy this then I give you this incentive), and not to the case where incentives are sunk costs (e.g., advertising dollars). Given this, the principal can in principle assign the entire budget to multiple arms if desired, in hopes that one of them is selected.

To see the power of effective incentives, note that incentives can sometimes induce an action to be selected forever that would otherwise never be selected. Consider a case with two actions, where initially the target action has value 2 and the non-target action has value 3. If either action is chosen, its value updates to 10. Assume $B = 2$. Without intervening in the first period the non-target action will be chosen, its value will update to 10, and it will be chosen forever even with incentives. However, by providing incentives on the target action in the first period it will be induced in that period and forever. The challenge is to design an incentive policy that is successful for all update models and even without knowledge of the update model. We consider two objective criteria.

3.1 Induce once

Consider a principal who wishes to induce action g once as soon as possible by providing effective incentives.

Problem 1 (Induce-Once). *For a given instance and a budget B, provide incentives to minimize the time t such that $x_g(t) = x_g^1$.*

If a solution does not exist, the minimum is infinity. Note that for action g to be selected at time t it is necessary that $B \geq \max_{i \in K_{-g}} [v_i(x_i(t)) - v_g(x_g(t))]$, at which point it is sufficient to provide $\Delta_g(t) = B$. The Induce-Once problem is thus identical to finding incentives that most quickly lead the values of all other actions to drop below the *inducible threshold* $T_{\text{once}} = B + v_g(x_g^0)$. For any threshold value T, we define the following:

Definition 1. *A threshold T for inducing action g is met at time t if and only if $v_i(x_i(t)) \leq T$ for all $i \in K_{-g}$.*

At first glance, it may appear that providing incentives to actions other than the target action g can be beneficial, by leading an action with value higher than the threshold to be selected and subsequently significantly drop in value, and in particular, to below the inducible threshold. This intuition turns out to be wrong! Any action above the inducible threshold will in any case be selected by the agent before action g until its value drops below the threshold, even without intervention. Getting such an action to be selected more quickly is possible through incentives, but this does not lead to action g being selected any sooner.

[3] For simplicity of exposition, we assume that the agent breaks ties in favor of the target action when there is a tie but otherwise in an arbitrary way. We can replace this assumption, which favors the target action, with any other tie-breaking rule, and all our results would continue to hold.

We formalize this observation as the 'threshold lemma,' which we will apply throughout this paper.

LEMMA 1 (THRESHOLD LEMMA). *Given a threshold T, let $k_i = \min\{k : v_i(x_i^k) \leq T\}$, for all $i \in K_{-g}$. Assume such k_i exist. Any incentive policy Δ that assigns $\Delta_i(t) = 0$ for all $i \in K_{-g}$ and $\Delta_g(t) \geq 0$ at every time t has the following properties:*

(a) At any time t before the threshold is first met, $x_i(t) = x_i^{m_i}$ satisfies $m_i \leq k_i$ for all $i \in K_{-g}$.

(b) If the threshold is first met at time t, then $x_i(t) = x_i^{k_i}$ for all $i \in K_{-g}$.

PROOF. Consider part (a). It suffices to show that at any time t before the threshold is first met, any action $i \in K_{-g}$ with $x_i(t) = x_i^{k_i}$ would not be selected at time t. Since the threshold is not yet met at such a time t, there exists $j \in K_{-g}$ such that $j \neq i$ and $v_j(x_j(t)) > T$. Under Δ, action i would not be selected at time t because $v_i(x_i(t)) + \Delta_i(t) = v_i(x_i^{k_i}) \leq T < v_j(x_j(t)) = v_j(x_j(t)) + \Delta_j(t)$, and so action j is strictly preferred.

Now consider part (b). If the threshold is first met at time t then exactly one action, say $\ell \in K_{-g}$, had been selected $k_\ell - 1$ times by period $t - 1$ and was selected in period $t - 1$ and every other action $j \in K_{-g}, j \neq \ell$ had already been selected at least k_j times by period $t - 1$. By (a), these other actions had been selected exactly k_j times by period $t - 1$ and hence $x_i(t) = x_i^{k_i}$ for all $i \in K_{-g}$ in period t. □

The threshold lemma shows that only providing incentives to the target action ensures that no other action is selected more times than needed before the threshold is met. Note that it does not guarantee the threshold will be met; that still needs to be shown for a particular incentive policy and corresponding threshold.

We next introduce a simple incentive policy that is central in our analysis. Its acronym hints at its guarantees.

DEFINITION 2. *The 'only provide to target' (OPT) incentive policy assigns $\Delta_g(t) = B$ and $\Delta_i(t) = 0$ for all $i \in K_{-g}$ for every time t.*

Note that in defining OPT we did not make any assumptions regarding its knowledge of current values or future updates.

THEOREM 1. *In the online model and under a per-period budget, OPT always provides the optimal offline solution to* INDUCE-ONCE.

PROOF. Consider $T_{\text{once}} = v_g(x_g^0) + B$ and define $k_i = \min\{k : v_i(x_i^k) \leq T_{\text{once}}\}$ for all $i \in K_{-g}$, and consider the interesting case in which this exists for every action so that a solution is not trivially precluded. The best possible solution will induce the agent to select the goal action after the necessary k_i activations of each action $i \in K_{-g}$. But actions $i \in K_{-g}$ can be selected no more than k_i times before the threshold is met by part (a) of the threshold lemma, and thus the threshold must be met under OPT. By applying part (b) of the threshold lemma, OPT makes the fewest selections of actions in K_{-g} necessary to meet the threshold, plus an additional necessary step to induce the target action. □

The key observation is that nothing the principal can do will speed up the agent's exploration of currently better actions. The principal can do worse than OPT however, e.g., by placing incentives on an action other than the target whose value is below the threshold and whose value in the state transitioned to is much higher.

3.2 Induce multiple times

In the motivating examples we consider, the principal may want the agent to select the target action (e.g. check answers, use a particular technique, or buy a product) more than once. This leads to the next objective criterion.

PROBLEM 2 (INDUCE-MULTI). *For a given instance, a budget B, and a number of rounds R, provide incentives to maximize m such that $x_g(R) = x_g^m$.*

Let us first tackle the related problem of minimizing the time to get m selections, for a given m. We know from Theorem 1 that OPT is the optimal incentive policy for $m = 1$. Furthermore, for $m \geq 2$, we know that OPT gets each subsequent selection of action g most quickly from *any* state configuration. However, this is not enough to conclude that OPT is the optimal incentive policy for getting m selections, because there may be other incentive policies that are slower than OPT at getting the first selection but faster in getting subsequent selections. While such incentive policies exist, we use the threshold lemma to show that they can do no better than OPT in minimizing the total amount of time needed to get m selections:

LEMMA 2. *In the online model and under a per-period budget, and for any fixed $m > 1$, OPT minimizes the time t such that $x_g(t) = x_g^m$.*

PROOF. Let $w = \text{argmin}_{0 \leq \ell < m} v_g(x_g^i)$ and let $T_{\text{multi}} = v_g(x_g^w) + B$. Let $k_i = \min\{k : v_i(x_i^k) \leq T_{\text{multi}}\}$ for all $i \in K_{-g}$, and consider the case in which this exists for every action so that a solution is not trivially precluded. The best possible solution will induce the agent to select the goal action the m-th time after the necessary k_i activations to each action $i \in K_{-g}$. But actions $i \in K_{-g}$ can be selected no more than k_i times before the threshold is met by part (a) of the threshold lemma, and thus the threshold must be met under OPT. Consider the period in which the threshold is first met. By applying part (b) of the threshold lemma, OPT makes the fewest selections of actions in K_{-g} necessary to meet the threshold, and since only the target item is selected thereafter until m selections are made, this completes the proof. □

By defining the threshold as the minimum value attained by action g before m selections we can apply the same idea as in the proof of Theorem 1. A fixed number of selections must necessarily occur on the other actions, and once they occur under OPT these actions will no longer be selected again.

THEOREM 2. *In the online model and under a per-period budget, OPT always provides the optimal offline solution to* INDUCE-MULTI.

PROOF. Let m denote the number of selections of the target action in time R under OPT. Assume for contradiction that there exist an incentive policy to induce the target $m' > m$ times in R steps. But by Lemma 2, OPT must also be able to induce the target action m' times in the same number or fewer time steps. This is a contradiction. □

4. FIXED ACROSS-PERIOD BUDGET

In this section, we consider a setting in which the principal has a budget that is fixed over time, and must decide on how to allocate that budget across time in order to induce the target action g once or multiple times. Formally we define the budget constraint on Δ as $\sum_{t=1}^{\infty} \Delta_{i(t)}(t) \leq B$, where $i(t)$ denotes the agent's selection at time t. We still require that incentives are non-negative, i.e., $\Delta_i(t) \geq 0$ for all actions i and times t.

This problem seems more difficult than the per-period budget problem because the principal must now decide how to split its budget across rounds. Providing too little in a particular round can miss an opportunity given the current state, whereas providing too much may make it difficult to induce future selections of the target action. As we will show, this turns out to be a nonissue if we wish to induce the target action once, but prevents any online algorithm from providing performance guarantees if we wish to induce the target action multiple times.

4.1 Induce once

We first return to INDUCE-ONCE, that is, we have a principal who wishes to induce action g once and as soon as possible. However, now the incentive policies under consideration have a fixed budget B across time.

Consider using OPT for this problem. OPT is optimal for the per-period budget case when B is available each period. Moreover, OPT in fact spends no money when the target is not selected, and so remains feasible even for a fixed budget across rounds and therefore optimal for this more constrained problem. The proof of this theorem is omitted as it is essentially identical to the proof of Theorem 1.

THEOREM 3. *In the online model and under a fixed across-period budget, OPT always provides the optimal offline solution to* INDUCE-ONCE.

4.2 Induce multiple times

Now consider the INDUCE-MULTI problem with a principal who wishes to induce the target action as many times as possible in a fixed number of rounds R. OPT is no longer optimal here because it may be beneficial to split the budget with the aim of getting more selections of the target action.

Consider a setting with two actions and a total budget of $B = 1$. Action 1 is the goal action. It may be that $v_2(x_2^0) = v_g(x_g^0) + B$ and v_2 increases in future states. By providing B on action g in the first period the goal is induced once, compared to zero successes with any other policy. On the other hand, suppose instead that $v_2(x_2^0) = v_g(x_g^0) + \epsilon$, some $0 < \epsilon < B$, and the value of both actions remains constant in all states. By providing B on g in the first period only one activation is achieved, whereas $\min(R, 1/\epsilon)$ could be achieved by providing ϵ on action g while budget remains (and this can be made arbitrarily large by increasing R and decreasing ϵ.)

In the online algorithms literature, an online algorithm for a maximization problem is α-competitive if the ratio between the optimal *offline* solution and the algorithm's solution is at most α for any given instance. Theorems 1 and 2 can be reformulated to state that under a per-period budget OPT is 1-competitive for INDUCE-ONCE and INDUCE-MULTI, respectively. On the other hand, the above argument implies that under an across-period budget there is no deterministic online algorithm that provides a bounded competitive ratio for the INDUCE-MULTI problem.

Our next formal result strengthens the above observation; we show that even a *randomized* algorithm cannot achieve a bounded approximation ratio. When the algorithm is randomized, the 'game' is as follows: we choose a randomized algorithm, then the adversary chooses an input; the input chosen by the adversary does not depend on the realization of the algorithm's randomness. The theorem holds even if the algorithm is allowed to know the current values of the actions at each time! In other words, this impossibility holds even for algorithms that are significantly more powerful than those we considered earlier.

THEOREM 4. *Under a fixed across-period budget there is no randomized algorithm that provides a bounded competitive ratio for* INDUCE-MULTI, *even if the algorithm can see the current values of the actions.*

The proof appears in the appendix. This result implies that it will be important to consider empirical performance or average case analysis, for particular agent models, in order to make progress.

4.3 Offline problem

As a counterpoint to Theorem 4, we consider the offline case, in which the principal knows the agent's value for any state of the actions the agent may reach and that state transitions are deterministic. This corresponds to a situation in which the agent is of known design, and that the principal has full understanding of the dynamics within the agent's decision environment.

The question of interest is whether there exists a tractable solution to this problem. An effective incentive policy would need to figure out when to provide incentives and how to split the budget across time periods, and a brute force computation of the optimal incentive to provide at each time step is too expensive.

THEOREM 5. *In the offline model and under a fixed across-period budget, an optimal solution to* INDUCE-MULTI *can be found in polynomial time.*

The proof involves the analysis of a nontrivial incentive policy; we give the outline here and relegate the proof of the key lemma to the appendix. To break down the problem, we first consider finding fixed budget incentive policies to solve the following subproblem.

PROBLEM 3. *Given $\bar{t} > 1$ and $m > 1$, and a budget B, find an incentive policy such that $x_g(\bar{t}) = x_g^\ell$ for $\ell \geq m$ when a solution exists.*

Essentially, if we can find an incentive policy that can get at least m selections in $\bar{t}$ rounds whenever possible for any $m > 1$, we can fix $\bar{t} = R$ and do a binary search over m to solve INDUCE-MULTI.

To get m selections within $\bar{t}$ time steps it is necessary that the agent selects the non-target actions no more than $\bar{t} - m$ times. An effective incentive policy should provide incentives on the target action when the other actions are least desirable, regardless of the value of the target action. This is the state in which it is cheapest to activate the target action.

We define the relevant activation threshold by simulating the agent function on actions K_{-g} only for $\bar{t} - m$ periods with no incentives, and computing

$$\underline{v} = \min_{1 \leq t \leq \bar{t}+1-m} \max_{i \in K_{-g}} v_i(x_i(t)). \tag{2}$$

It is easy to see that this threshold, $\underline{v}$, can be computed in polynomial time.

Definition 3. *The 'only provide to target when cheap' (OPTc) incentive policy assigns $\Delta_i(t) = 0$ for all $i \in K$ until the threshold $T = \underline{v}$ is met, where $\underline{v}$ is defined as in Equation* (2)*. Let t' denote the period in which the threshold is first met. OPTc provides $\Delta_g(t) = \max\{0, \underline{v} - v_g(x_g(t))\}$ for $t \geq t'$ while there is enough budget remaining and $\Delta_g(t) = 0$ otherwise. No incentives are ever provided to actions in K_{-g}.*

Now, by using binary search, the following lemma is sufficient to prove Theorem 5.

Lemma 3. *In the offline model and under a fixed across-period budget, OPTc solves Problem 3.*

The proof of the lemma appears in the appendix. An interesting aspect of OPTc is that it has much of the same structure as OPT: the optimal incentive policy does not modify the agent's learning process until the point where the agent can best be incentivized to select the target action. The offline problem remains a problem about when to provide incentives on the target action and not how to intervene in the selection process on other actions.

4.4 Case study: induce a new action

Theorem 4 and Theorem 5 establish strong negative and positive results at opposing ends of the information spectrum. In most realistic scenarios we expect the principal to have some (but not full) knowledge of the agent's update process and reward distribution. Sticking with an across-period budget and the objective of maximizing the number of selections of the target arms within a fixed number of rounds, we demonstrate how to utilize such knowledge to find effective incentives for a particular problem of interest.

Consider a scenario in which there are two actions, one whose value to the agent is fixed and another whose realized rewards are drawn from a stationary distribution known to the principal. Moreoever, assume the agent's belief updates based on the empirical average of rewards, and is initialized by a draw from the same distribution (e.g., the agent gets an initial sample). This scenario models an agent's choice between an incumbent option (e.g., a known product, service, or technique) and a new entrant option, where the principal wishes to entice the agent toward the new option by providing appropriate incentives.

To be more concrete, let us consider a particular instance of the problem. There are 2 rounds. The value of the incumbent option is fixed at 1, and the reward from selecting the new option and his initial belief on its value are sampled uniformly from 0 to 1. The process reflects uncertainty about the new option's quality, and the agent's updating beliefs reflect his estimate of the new option's value. The principal has a budget of 1 to be spread across the rounds, and can observe which action the agent selects but not the realized rewards. The goal is to maximize the expected number of selections of the new action.

In analyzing this problem, note that the agent's decision in the first period provides some information on the agent's value at the time, and thus, the distribution of possible values following the update. Furthermore, if the agent does not select the new action in the first period, then no money is spent and we can guarantee a selection in the second period. Solving for the optimal incentive policy analytically, we get the following fact:

Fact 1. *The optimal incentive policy for this problem provides $\frac{4}{9}$ to the new action in the first period and the remaining budget in the second period. The expected number of selections is $\frac{25}{18}$.*

Proof (sketch). Let T_i represent an indicator variable over whether the agent selects the new action in period i, such that $P(T_i)$ represents the probability of the selection given the principal's uncertainty over the agent's current value for the action and the provided incentive. Let α represent the incentive provided to the new action in the first period. We wish to maximize the expected number of selections:

$$\begin{aligned} & 2P(T_1)P(T_2|T_1) + P(T_1)P(\neg T_2|T_1) + P(\neg T_1)P(T_2|\neg T_1) \\ = & \quad P(T_1)[P(T_2|T_1) + 1] + P(\neg T_1) \\ = & \quad P(T_1, T_2) + 1 \end{aligned}$$

Here $P(T_1, T_2) = P(1 - \alpha \leq r_0 \leq 1, \frac{r_0 + r_1}{2} \geq \alpha)$, where r_0 represents the initial value on the new action and r_1 represents the value from the first selection of the new action. By integrating the probabillity density function of the uniform distribution over the valid regions based on the value of α, we have:

$$P(T_1, T_2) = \begin{cases} \alpha & \text{if } \alpha \leq \frac{1}{3}. \\ -\frac{9}{2}\alpha^2 + 4\alpha - \frac{1}{2} & \text{if } \frac{1}{3} < \alpha \leq \frac{1}{2}. \\ -\frac{5}{2}\alpha^2 + 2\alpha & \text{if } \frac{1}{2} < \alpha \leq \frac{2}{3}. \\ 2\alpha^2 - 4\alpha + 2 & \text{if } \frac{2}{3} < \alpha \leq 1. \end{cases}$$

It is easy to check from here that the maximum is attained in the second segment, with $\alpha = 4/9$. □

Intuitively, providing too much incentives in the first period misses out on possible selections in the second period, and providing too little in the first period may likewise miss out on selections in the first period. Given this, 4/9 seems like a good choice for α. However, it is surprising that the optimal solution is not $\alpha = 1/2$: it turns out that by providing slightly less incentives in the first period, there is a higher chance (7/8 vs. 3/4) of getting selections in the second period conditional on a selection in the first period, because the value of the first draw must have been higher and more incentives are left over. Even though the principal is slightly less likely to get a selection in the first period, this helps to maximize the probability of getting a selection in both rounds, which we have shown is equivalent to maximizing the expected number of selections.

We can also consider the same problem but with 3 rounds, where the optimal incentive policy (obtained via simulation) provides about 0.37 in the first period, 0.27 in the second period if the new action is selected and 0.54 otherwise. It is interesting to note the disparity in the amount of incentives provided in the second period based on what happened in

	Induce-Once	Induce-Multi
Per-period	OPT optimal (Thm 1)	OPT optimal (Thm 2)
Fixed	OPT optimal (Thm 3)	Unbounded ratio (Thm 4)

Table 1: Summary of our results in the online model for the two objective criteria and different budget models for the principal.

the first period. Since success in the first period indicates a high draw in the first period and failure indicates a low draw, the observation serves as an informative signal about the amount of incentives required in the second period.

5. DISCUSSION

Table 1 summarizes our main results; rows correspond to the assumption made on the budget and columns to the optimization problem. The most striking aspect is the relation between the performance of OPT when one varies the assumption on the budget between per-period and fixed, and the problem between Induce-Once and Induce-Multi. As long as one of these dimensions remains fixed OPT is still optimal, but when we consider the harder variation in both dimensions then even a randomized policy that knows the current values cannot provide a bounded ratio!

Our incentive policy for the per-period budget case requires no knowledge of the agent's values or value update process, nor the number of repetitions for which we wish to induce the target, nor the time horizon over which the target is to be induced; it is optimal even with this knowledge. In this setting, it is not necessary for the principal to learn about the agent, for example by drawing inferences about the agent's values and update process from observed behavior of selected options. It is interesting, also, that the principal is unable to usefully perturb the agent's learning process until the point at which his desired goal action can be induced, and even if he knew the agent's values or value update process.

Interestingly, this is quite different in the across-period budget setting, where progress will require knowledge of an agent's selection and learning process or learning by the principal about the agent. The analytical approach demonstrated for finding incentives to induce an agent to select a new action uses both knowledge about the value update process and inference on the distribution of current values based on past decisions, and can be applied for rewards drawn from different distributions and for different update processes. Future work should seek to obtain tractable algorithms for finding effective incentives given a known model of agent behavior but private agent beliefs, and seek to gain a better understanding of the structure of effective incentive policies, on particular classes of problems.

In addition to variations on the budget constraint, one can consider variations to the agent's selection policy. For example, the agent might select the action with highest value with probability $1-\epsilon$, and select a random action with probability ϵ. It is possible to show that in this case OPT is no longer guaranteed to be optimal for a per-period budget, even with respect to Induce-Once.[4] It is also of interest to relax the assumption on the independence of actions, and to consider models with long-term learning in which the agent learns to internalize external incentives and change its own intrinsic value for future actions. Other objective criteria are also of interest, for example a principal that wants to induce an action followed by another action, in immediate succession.

Acknowledgments

The last author acknowledges support from the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. In addition, this work is supported in part by NSF grant CCF 0915016, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

6. REFERENCES

[1] M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. In *ACM-EC*, pages 79–88, 2009.

[2] D. Bergemann and J. Välimäki. The dynamic pivot mechanism. *Econometrica*, 78:771–789, 2010.

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

[4] R. I. Brafman and M. Tennenholtz. On partially controlled multi-agent systems. *JAIR*, 4:477–507, 1996.

[5] R. Cavallo, D. C. Parkes, and S. Singh. Optimal coordinated planning amongst self-interested agents with private state. In *UAI*, pages 55–62, 2006.

[6] R. D. Kleinberg. *Online decision problems with large strategy sets*. PhD thesis, MIT, 2005.

[7] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *K-CAP*, pages 9–16, 2009.

[8] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, pages 278–287, 1999.

[9] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[10] P. Stone and S. Kraus. To teach or not to teach? decision making under uncertainty in ad hoc teams. In *AAMAS*, 2010.

[11] H. Zhang, Y. Chen, and D. C. Parkes. A general approach to environment design with one agent. In *IJCAI*, pages 2002–2008, 2009.

[12] H. Zhang and D. C. Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI*, pages 208–214, 2008.

[13] H. Zhang, D. C. Parkes, and Y. Chen. Policy teaching through reward function learning. In *ACM-EC*, pages 295–304, 2009.

[4] Consider a case with three actions. The principal has a per-period budget of 2. Action 1 is the target and has its value fixed at 1. Action 2 is associated with the belief sequence $(v_2^0, v_2^1, \ldots) = (5, 0, 5, 5, \ldots)$, and action 3 is associated with the belief sequence $(v_3^0, v_3^1, \ldots) = (4, 0, 0, \ldots)$. Here action 1 can be induced by a non-random action if and only if action 2 is induced exactly one, and action 3 is induced at least once. For small ϵ, action 2 is most likely to be selected first under OPT. This is undesirable, however, since any random selection of action 2 henceforth will result in no future selections of action 1. It is better to instead provide incentives to action 3 in the first period (and apply OPT thereafter), so that we try to 'hold off' on selecting action 2 until the belief on the value of action 3 has dropped.

APPENDIX

A. PROOFS

A.1 Proof of Theorem 4

We assume without loss of generality that the budget size is 1. Let $k \in \mathbb{N}$; assume for contradiction that there is a randomized online algorithm with a competitive ratio α (worst-case ratio of number of activations of g in offline optimal to expected number of activations of g in online algorithm over all instances) smaller than k. Set $\epsilon = 1/(10k)$.

We consider a setting where there are just two actions, the target action g and the non-target h. We design an infinite family of inputs $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_j, \ldots$ Note that an input simply specifies a number of rounds, and a sequence of values for g and h. For all $j \in \mathbb{N} \cup \{0\}$, the sequence of values that $\mathcal{I}_j$ assigns to g is all zeros, that is, $v_g(x_g^t) = 0$ for all t; the inputs only differ on their values h. The sequence of values $v_h(x_h^0), v_h(x_h^1), \ldots$ that $\mathcal{I}_j$ assigns to this action is

$$1, \epsilon, \epsilon^2, \ldots, \epsilon^j, 2, 2, \ldots, 2, \ldots$$

We do not specify the number of rounds, as we can choose it to be large enough for it not to be an issue.

Given a run of the algorithm on some input $\mathcal{I}_j$, we refer to the sequence of selections of action g while action h has a value ϵ^p as *phase p*. Once h is selected we move to phase $p+1$. Note that for each select of g in phase p the algorithm has to invest ϵ^p of its budget.

Let Z_p^j be a random variable that denotes the budget spent by the algorithm within phase p given the input $\mathcal{I}_j$, where the randomness comes from the algorithm's coin flips. The crux of the proof is the following lemma.

LEMMA 4. *For every $j \in \mathbb{N} \cup \{0\}$ and every $p \in \{0, \ldots, j\}$, if the randomized online algorithm has competitive ratio smaller than k then $\mathbb{E}[Z_p^j] \geq \epsilon$.*

PROOF. Assume for contradiction that this is not the case, i.e., there is some $j \in \mathbb{N} \cup \{0\}$ and $p \in \{0, \ldots, j\}$ such that $\mathbb{E}[Z_p^j] < \epsilon$. Up to phase p the algorithm cannot distinguish between $\mathcal{I}_j$ and $\mathcal{I}_p$ (due to the online nature of the model), hence it holds that $\mathbb{E}[Z_p^p] < \epsilon$, that is, the algorithm spends less than ϵ in expectation in phase p given the input $\mathcal{I}_p$. It follows that the expected number of times g is selected in phase p is smaller than $\epsilon/\epsilon^p = 10^{p-1}k^{p-1}$. Given $\mathcal{I}_p$, the algorithm will no longer be able to select g after phase p (since then the value of h is then 2). We derive an upper bound on the expected number of times the algorithm selects g on $\mathcal{I}_p$ by generously allowing the algorithm spend a budget of 1 in every phase $p' < p$. The upper bound is then

$$1 + 10k + \cdots + 10^{p-1}k^{p-1} + 10^{p-1}k^{p-1} \leq 3 \cdot 10^{p-1}k^{p-1}.$$

On the other hand, the optimal offline solution on $\mathcal{I}_p$ selects g $10^p k^p$ times, i.e., the ratio α is at least $(10/3)k$, in contradiction to the assumption that the algorithm's competitive ratio is smaller than k. □

Now, consider input $\mathcal{I}_{j^*}$ for some $j^* \in \mathbb{N}, j^* > 1/\epsilon$. By Lemma 4 we have that $\mathbb{E}[Z_p^{j^*}] \geq \epsilon$ for all $p \in \{0, \ldots, j^*\}$. It follows from the linearity of expectation that

$$\mathbb{E}\left[\sum_{p=0}^{j^*} Z_p^{j^*}\right] = \sum_{p=0}^{j^*} \mathbb{E}[Z_p^{j^*}] > \frac{1}{\epsilon} \cdot \epsilon = 1. \tag{3}$$

However, since the total budget size is 1 the random variable $\sum_{p=0}^{j^*} Z_p^{j^*}$ must take values in $[0, 1]$, and in particular $\mathbb{E}[\sum_{p=0}^{j^*} Z_p^{j^*}] \leq 1$. This is a contradiction to Equation (3). □

A.2 Proof of Lemma 3

We first establish that $\underline{v}$, as defined in Equation (2), is the threshold where it is cheapest to provide incentives to the target over at most $\bar{t} - m$ periods of selecting actions other than g. For this, let

$$v^* = \min_{m_{-g}} \max_{i \in K_{-g}} v_i(x_i^{m_i}) \tag{4}$$
$$\text{s.t.} \quad \sum_{i \in K_{-g}} m_i \leq \bar{t} - m$$

represent the lowest value of the highest action with up to $\bar{t} - m$ selections of the non-target actions, and where $m_{-g} = (m_1, \ldots, m_{g-1}, m_{g+1}, \ldots, m_n)$.

We want to establish that $\underline{v} = v^*$. Indeed, clearly $v^* \leq \underline{v}$ by definition. Suppose for contradiction that $\underline{v} > v^*$. Let m_{-g}^* minimize the expression in Eq. (4). Consider running the simulation used to define $\underline{v}$ for $\sum_{i \in K_{-g}} m_i^*$ rounds, and let ℓ_i denote the number of times that each action $i \in K_{-g}$ is selected in this process.

If $\ell_i = m_i^*$ for all $i \in K_{-g}$ then clearly $\underline{v} = v^*$, since it is attained in the final period of the simulation, and this is a contradiction. Otherwise, and using the fact that $\sum_{i \in K_{-g}} \ell_i = \sum_{i \in K_{-g}} m_i^*$, then there exists some $j \in K_{-g}$ such that $m_j^* < \ell_j$. Note that,

$$v_j(x_j^{m_j^*}) \leq v^* < \underline{v}, \tag{5}$$

where the first inequality holds by definition of v^* and the second by assumption. Now there is some time t during the simulation where $x_j(t) = x_j^{m_j^*}$, and action j is selected. But by definition of $\underline{v}$ the value of the action that is selected by the agent must be at least $\underline{v}$, in contradiction to (5). This establishes $\underline{v} = v^*$.

In order to complete the proof of the lemma, we now know that OPTc uses $\underline{v} = v^*$ as the threshold T. Let $k_i = \min\{k : v_i(x_i^k) \leq T\}$ for all $i \in K_{-g}$. By definition of v^*, $\sum_{i \in K_{-g}} k_i \leq \bar{t} - m$. Note that OPTc satisfies the conditions of the threshold lemma. Proceed by case analysis. If the threshold is not met after $\bar{t}$ rounds then, by part (a) of the threshold lemma, action g must have been selected at least m times and the case is established. Otherwise, if the threshold is met, it is met after at most $\bar{t} - m$ selections of actions in K_{-g} by part (b) of the threshold lemma. For any incentive policy to get m selections in $\bar{t}$ rounds, it must have provided at least $\max\{0, v^* - v_g(x_g^\ell)\}$ to get selection number $\ell + 1$ of action g, for each of $\ell \in \{0, 1, \ldots, m-1\}$. Since OPTc spends no budget before the threshold is met and once it is met it provides exactly $\max\{0, v^* - v_g(x_g^\ell)\}$ for selection number $\ell + 1$ of action g, for each of $\ell \in \{0, 1, \ldots, k-1\}$, then OPTc will get at least m selections of action g whenever this is possible under any incentive policy. This completes the case, and the proof. □