# The Markovian Price of Information

Anupam Gupta[1], Haotian Jiang[2] ✉, Ziv Scully[1], and Sahil Singla[3]

[1] Carnegie Mellon University, Pittsburgh PA 15213, USA
{anupamg,zscully}@cs.cmu.edu
[2] University of Washington, Seattle WA 98195, USA
jhtdavid@uw.edu
[3] Princeton University, Princeton NJ 08544, USA
singla@cs.princeton.edu

**Abstract.** Suppose there are $n$ Markov chains and we need to pay a per-step *price* to advance them. The "destination" states of the Markov chains contain rewards; however, we can only get rewards for a subset of them that satisfy a combinatorial constraint, e.g., at most $k$ of them, or they are acyclic in an underlying graph. What strategy should we choose to advance the Markov chains if our goal is to maximize the total reward *minus* the total price that we pay?

In this paper we introduce a Markovian price of information model to capture settings such as the above, where the input parameters of a combinatorial optimization problem are given via Markov chains. We design optimal/approximation algorithms that jointly optimize the value of the combinatorial problem and the total paid price. We also study *robustness* of our algorithms to the distribution parameters and how to handle the *commitment* constraint.

Our work brings together two classical lines of investigation: getting optimal strategies for Markovian multi-armed bandits, and getting exact and approximation algorithms for discrete optimization problems using combinatorial as well as linear-programming relaxation ideas.

**Keywords:** Multi-armed bandits · Gittins index · Probing algorithms.

## 1 Introduction

Suppose we are running an oil company and are deciding where to set up new drilling operations. There are several candidate sites, but the value of drilling each site is a random variable. We must therefore *inspect* sites before drilling. Each inspection gives more information about a site's value, but the inspection process is costly. Based on laws, geography, or availability of equipment, there are constraints on which sets of drilling sites are feasible. We ask:

> What adaptive inspection strategy should we adopt to find a feasible set of sites to drill which maximizes, in expectation, the value of the chosen (drilled) sites minus the total inspection cost of all sites?

Let us consider the optimization challenges in this problem:

(i) Even if we could fully inspect each site for free, choosing the best feasible set of sites is a *combinatorial optimization* problem.

(ii) Each site may have *multiple stages* of inspection. The costs and possible outcomes of later stages may depend on the outcomes of earlier stages. We use a *Markov chain* for each site to model how our knowledge about the value of the site stochastically evolves with each inspection.

(iii) Since a site's Markov chain model may not exactly match reality, we want a *robust* strategy that performs well even under small changes in the model parameters.

(iv) If there is competition among several companies, it may not be possible to do a few stages of inspection at a given site, abandon that site's inspection to inspect other sites, and then later return to further inspect the first site. In this case the problem has additional "take it or leave it" or *commitment* constraints, which prevent interleaving inspection of multiple sites.

While each of the above aspects has been individually studied in the past, no prior work addresses all of them. In particular, aspects (i) and (ii) have not been simultaneously studied before. In this work we advance the state of the art by solving the (i)-(ii)-(iii) and the (i)-(ii)-(iv) problems.

To study aspects (i) and (ii) together, in §2 we propose the *Markovian Price of Information* (Markovian PoI) model. The Markovian PoI model unifies prior models which address (i) or (ii) alone. These prior models include those of Kleinberg et al. [17] and Singla [18], who study the combinatorial optimization aspect (i) in the so-called *price of information* model, in which each site has just a single stage of inspection; and those of Dimitriu et al. [8] and Kleinberg et al. [17, Appendix G], who consider the multiple stage inspection aspect (ii) for the problem of selecting just a single site.

Our main results[4] show how to solve combinatorial optimization problems, including both maximization and minimization problems, in the Markovian PoI model. We give two methods of transforming classic algorithms, originally designed for the Free-Info (inspection is free) setting, into *adaptive* algorithms for the Markovian PoI setting. These adaptive algorithms respond dynamically to the random outcomes of inspection.

- In §3.3 we transform "greedy" $\alpha$-approximation algorithms in the Free-Info setting into $\alpha$-approximation adaptive algorithms in the Markovian PoI setting (Theorem 1). For example, this yields optimal algorithms for matroid optimization (Corollary 1).
- In §4 we show how to slightly modify our $\alpha$-approximations for the Markovian PoI setting in Theorem 1 to make them robust to small changes in the model parameters (Theorem 2).
- In §5 we use *online contention resolution schemes* (OCRSs) [10] to transform LP based Free-Info maximization algorithms into adaptive Markovian PoI algorithms while respecting the commitment constraints. Specifically, a $1/\alpha$-selectable OCRS yields $\alpha$-approximation with commitment (Theorem 3).

[4] Due to space constraints, we omit full proofs in this extended abstract. The full version is available at https://arxiv.org/abs/1902.07856.

The general idea behind our first result (Theorem 1) is the following. A Frugal combinatorial algorithm (Definition 8) is, roughly speaking, “greedy”: it repeatedly selects the feasible item of greatest marginal value. We show how to adapt *any* Frugal algorithm to the Markovian PoI setting:

- Instead of using a fixed value for each item $i$, we use a *time-varying “proxy” value* that depends on the state of $i$’s Markov chain.
- Instead of immediately selecting the item $i$ of greatest marginal value, we *advance $i$’s Markov chain one step.*

The main difficulty lies in choosing each item’s proxy value, for which simple heuristics can be suboptimal. We use a quantity for each state of each item’s Markov chain called its *grade*, and an item’s proxy value is its *minimum grade so far.* A state’s grade is closely related to the Gittins index from the multi-armed bandit literature, which we discuss along with other related work in §6.

## 2 The Markovian Price of Information Model

To capture the evolution of our knowledge about an item’s value, we use the notion of a Markov system from [8] (who did not consider values at the destinations).

**Definition 1 (Markov System).** *A Markov system $\mathcal{S} = (V, P, s, T, \boldsymbol{\pi}, \mathbf{r})$ for an element consists of a discrete Markov chain with state space $V$, a transition matrix $P = \{p_{u,v}\}$ indexed by $V \times V$ (here $p_{u,v}$ is the probability of transitioning from $u$ to $v$), a starting state $s$, a set of absorbing* destination *states $T \subseteq V$, a non-negative probing price $\pi^u \in \mathbb{R}_{\geq 0}$ for every state $u \in V \setminus T$, and a value $r^t \in \mathbb{R}$ for each destination state $t \in T$. We assume that every state $u \in V$ reaches some destination state.*

We have a collection $J$ of *ground elements*, each associated with its own Markov system. An element is *ready* if its Markov system has reached one of its absorbing destination states. For a ready element, if $\omega$ is the (random) *trajectory* of its Markov chain then $d(\omega)$ denotes its associated destination state. We now define the Markovian PoI game, which consists of an objective function on $J$.

**Definition 2 (Markovian PoI Game).** *Given a set of ground elements $J$, constraints $\mathcal{F} \subseteq 2^J$, an objective function $f : 2^J \times \mathbb{R}^{|J|} \to \mathbb{R}$, and a Markov system $\mathcal{S}_i = (V_i, P_i, s_i, T_i, \boldsymbol{\pi}_i, \mathbf{r}_i)$ for each element $i \in J$, the* Markovian PoI *game is the following. At each time step, we either advance a Markov system $\mathcal{S}_i$ from its current state $u \in V_i \setminus T_i$ by incurring price $\pi_i^u$, or we end the game by selecting a subset of* ready *elements $\mathbb{I} \subseteq J$ that are* feasible*—i.e., $\mathbb{I} \in \mathcal{F}$.*

A common choice for $f$ is the *additive* objective $f(\mathbb{I}, \mathbf{x}) = \sum_{i \in \mathbb{I}} x_i$.

Let $\boldsymbol{\omega}$ denote the *trajectory profile* for the Markovian PoI game: it consists of the random trajectories $\omega_i$ taken by all the Markov chains $i$ at the end of the game. To avoid confusion, we write the selected feasible solution $\mathbb{I}$ as $\mathbb{I}(\boldsymbol{\omega})$. A utility/disutility optimization problem is to give a strategy for a Markovian PoI game while optimizing both the objective and the total price.

**Utility Maximization** (Util-Max): A Markovian PoI game where the constraints $\mathcal{F}$ are *downward-closed* (i.e., *packing*) and the values $\mathbf{r}_i$ are non-negative for every $i \in J$ (i.e., $\forall t \in T_i$, $r_i^t \geq 0$, and can be understood as a reward obtained for selecting $i$). The goal is to find a strategy ALG maximizing *utility*:

$$U^{\max}(\mathrm{ALG}) \triangleq \mathbb{E}_{\boldsymbol{\omega}}\Big[\underbrace{f\left(\mathbb{I}(\boldsymbol{\omega}), \{r_i^{d(\omega_i)}\}_{i\in\mathbb{I}(\boldsymbol{\omega})}\right)}_{\text{value}} - \underbrace{\textstyle\sum_i \sum_{u\in\omega_i} \pi_i^u}_{\text{total price}}\Big]. \qquad (1)$$

Since the empty set is always feasible, the optimum utility is non-negative.

We also define a minimization variant of the problem that is useful to capture covering combinatorial problems such as minimum spanning trees and set cover.

**Disutility Minimization** (Disutil-Min) : A Markovian PoI game where the constraints $\mathcal{F}$ are *upward-closed* (i.e., *covering*) and the values $\mathbf{r}_i$ are non-negative for every $i \in J$ (i.e., $\forall t \in T_i$, $r_i^t \geq 0$, and can be understood as a cost we pay for selecting $i$). The goal is to find a strategy ALG minimizing *disutility*:

$$U^{\min}(\mathrm{ALG}) \triangleq \mathbb{E}_{\boldsymbol{\omega}}\Big[f\left(\mathbb{I}(\boldsymbol{\omega}), \{r_i^{d(\omega_i)}\}_{i\in\mathbb{I}(\boldsymbol{\omega})}\right) + \textstyle\sum_i \sum_{u\in\omega_i} \pi_i^u\Big].$$

We will assume that the function $f$ is non-negative when all $\mathbf{r}_i$ are non-negative. Hence, the disutility of the optimal policy is non-negative.

In the special case where all the Markov chains for a Markovian PoI game are formed by a *directed acyclic graph* (Dag), we call the corresponding optimization problem Dag-Util-Max or Dag-Disutil-Min.

## 3 Adaptive Utility Maximization via Frugal Algorithms

Frugal algorithms, introduced in Singla [18], capture the intuitive notion of "greedy" algorithms. There are many known Frugal algorithms, e.g., optimal algorithms for matroids and $O(1)$-approx algorithms for matchings, vertex cover, and facility location. These Frugal algorithms were designed in the traditional *free information* (Free-Info) setting, where each ground element has a fixed value. Can we use them in the Markovian PoI world?

Our main contribution is a technique that adapts *any* Frugal algorithm to the Markovian PoI world, achieving the *same approximation ratio* as the original algorithm. The result applies to *semiadditive* objective functions $f$, which are those of the form $f(\mathbb{I}, \mathbf{x}) = \sum_{i\in\mathbb{I}} x_i + h(\mathbb{I})$ for some $h : 2^J \to \mathbb{R}$.

**Theorem 1.** *For a semiadditive objective function* $\mathsf{val}$*, if there exists an* $\alpha$*-approximation* Frugal *algorithm for a* Util-Max *problem over some packing constraints* $\mathcal{F}$ *in the* Free-Info *world, then there exists an* $\alpha$*-approximation strategy for the corresponding* Util-Max *problem in the* Markovian PoI *world.*

We prove an analogous result for Disutil-Min in the full version. The following corollaries immediately follow from known Frugal algorithms [18].

**Corollary 1.** *In the* Markovian PoI *world, we have:*

- *An optimal algorithm for both* Util-Max *and* Disutil-Min *for matroids.*
- *A* 2*-approx for* Util-Max *for matchings and a $k$-approx for a $k$-system.*
- *A* $\min\{\theta, \log n\}$*-approx for* Disutil-Min *for set-cover, where $\theta$ is the maximum number of sets in which a ground element is present.*
- *A* 1.861*-approx for* Disutil-Min *for facility location.*
- *A* 3*-approx for* Disutil-Min *for prize-collecting Steiner tree.*

Before proving Theorem 1, we define a *grade* for every state in a Markov system in §3.1, much as in [8]. This grade is a variant of the popular *Gittins index.* In §3.2, we use the grade to define a *prevailing cost* and an *epoch* for a trajectory. In §3.3, we use these definitions to prove Theorem 1. We consider Util-Max throughout, but analogous definitions and arguments hold for Disutil-Min.

### 3.1 Grade of a State

To define the *grade* $\tau^v$ of a state $v \in V$ in Markov system $\mathcal{S} = (V, P, s, T, \boldsymbol{\pi}, \mathbf{r})$, we consider the following Markov game called $\tau$*-penalized* $\mathcal{S}$, denoted $\mathcal{S}(\tau)$. Roughly, $\mathcal{S}(\tau)$ is the same as $\mathcal{S}$ but with a *termination penalty*, which is a constant $\tau \in \mathbb{R}$.

Suppose $v \in V$ denotes the current state of $\mathcal{S}$ in the game $\mathcal{S}(\tau)$. In each move, the player has two choices: (a) *Halt* that immediately ends the game, and (b) *Play* that changes the state, price, and value as follows:

- If $v \in V \setminus T$, the player pays price $\pi^v$, the current state of $\mathcal{S}$ changes according to the transition matrix $P$, and the game continues.
- If $v \in T$, then the player receives *penalized value* $r^v - \tau$, where $\tau$ is the aforementioned termination penalty, and the game ends.

The player wishes to maximize his *utility*, which is the expected value he obtains minus the expected price he pays. We write $U^v(\tau)$ for the utility attained by optimal play starting from state $v \in V$.

The utility $U^v(\tau)$ is clearly non-increasing in the penalty $\tau$, and one can also show that it is continuous [8, Section 4]. In the case of large penalty $\tau \to +\infty$, it is optimal to halt immediately, achieving $U^v(\tau) = 0$. In the opposite extreme $\tau \to -\infty$, it is optimal to play until completion, achieving $U^v(\tau) \to +\infty$. Thus, as we increase $\tau$ from $-\infty$ to $+\infty$, the utility $U^v(\tau)$ becomes 0 at some critical value $\tau = \tau^v$. This critical value $\tau^v$ that depends on state $v$ is the *grade.*

**Definition 3 (Grade).** *The* grade *of a state $v$ in Markov system $\mathcal{S}$ is* $\tau^v \triangleq \sup\{\tau \in \mathbb{R} \mid U^v(\tau) > 0\}$. *For a* Util-Max *problem, we write the grade of a state $v$ in Markov system $\mathcal{S}_i$ corresponding to element $i$ as $\tau_i^v$.*

The quantity grade of a state is well-defined from the above discussion. We emphasize that it is independent of all other Markov systems. Put another way, the grade of a state is the penalty $\tau$ that makes the player *indifferent* between halting and playing. It is known how to compute grade efficiently [8, Section 7].

### 3.2 Prevailing Cost and Epoch

We now define a *prevailing cost* [8] and an *epoch*. The prevailing cost of Markov system $\mathcal{S}$ is its minimum grade at any point in time.

**Definition 4 (Prevailing Cost).** *The* prevailing cost *of Markov system $\mathcal{S}_i$ in a trajectory $\omega_i$ is $Y^{\max}(\omega_i) = \min_{v \in \omega_i}\{\tau_i^v\}$. For trajectory profile $\boldsymbol{\omega}$, denote $Y^{\max}(\boldsymbol{\omega})$ the list of prevailing costs for each Markov system.*

Put another way, the prevailing cost is the maximum termination penalty for the game $\mathcal{S}(\tau)$ such that for every state along $\omega$ the player does not want to halt.

Observe that the prevailing cost of a trajectory can only decrease as it extends further. In particular, it decreases whenever the Markov system reaches a state with grade smaller than each of the previously visited states. We can therefore view the prevailing cost as a non-increasing piecewise constant function of time. This motivates us to define an epoch.

**Definition 5 (Epoch).** *An* epoch *for a trajectory $\omega$ is any maximal continuous segment of $\omega$ where the prevailing cost does not change.*

Since the grade can be computed efficiently, we can also compute the prevailing cost and epochs of a trajectory efficiently.

### 3.3 Adaptive Algorithms for Utility Maximization

In this section, we prove Theorem 1 that adapts a Frugal algorithm in Free-Info world to a probing strategy in the Markovian PoI world. This theorem concerns *semiadditive functions*, which are useful to capture non-additive objectives of problems like facility location and prize-collecting Steiner tree.

**Definition 6 (Semiadditive Function [18]).** *A function $f(\mathbb{I}, \mathbf{X}) : 2^J \times \mathbb{R}^{|J|} \to \mathbb{R}$ is* semiadditive *if there exists a function $h : 2^J \to \mathbb{R}$ s.t. $f(\mathbb{I}, \mathbf{x}) = \sum_{i \in \mathbb{I}} x_i + h(\mathbb{I})$.*

All additive functions are semiadditive with $h(\mathbb{I}) = 0$ for all $\mathbb{I}$. To capture the facility location problem on a graph $G = (J, E)$ with metric $(J, d)$, clients $C \subseteq J$, and facility opening costs $\mathbf{x} : J \to \mathbb{R}_{\geq 0}$, we can define $h(\mathbb{I}) = \sum_{j \in C} \min_{i \in \mathbb{I}} d(j, i)$. Notice $h$ only depends on the identity of facilities $\mathbb{I}$ and not their opening costs.

The proof of Theorem 1 takes two steps. We first give a randomized reduction to upper bound the utility of the optimal strategy in the Markovian PoI world with the optimum of a *surrogate problem* in the Free-Info world. Then, we transform a Frugal algorithm into a strategy with utility close to this bound.

**Upper Bounding the Optimal Strategy Using a Surrogate.** The *main idea* in this section is to show that for Util-Max, no strategy (in particular, optimal) can derive more utility from an element $i \in J$ than its prevailing cost. Here, the prevailing cost of $i$ is for a random trajectory to a destination state in Markov system $\mathcal{S}_i$. Since the optimal strategy can only select a feasible set in $\mathcal{F}$, this idea naturally leads to the following FREE-INFO *surrogate problem*: imagine each element's value is exactly its (random) prevailing cost, the goal is to select a set feasible in $\mathcal{F}$ to maximize the total value. In Lemma 1, we show that the expected optimum value of this surrogate problem is an upper bound on the optimum utility for UTIL-MAX. First, we formally define the surrogate problem.

**Definition 7 (Surrogate Problem).** *Given a* UTIL-MAX *problem with semi-additive objective* $\mathsf{val}$ *and packing constraints* $\mathcal{F}$ *over universe* $J$*, the corresponding* surrogate *problem over* $J$ *is the following. It consists of constraints* $\mathcal{F}$ *and (random) objective function* $\tilde{f} : 2^J \to \mathbb{R}$ *given by* $\tilde{f}(\mathbb{I}) = \mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega}))$*, where* $\mathbf{Y}^{\max}(\boldsymbol{\omega})$ *denotes the prevailing costs over a random trajectory profile* $\boldsymbol{\omega}$ *consisting of independent random trajectories for each element* $i \in J$ *to a destination state. The goal is to select* $\mathbb{I} \in \mathcal{F}$ *to maximize* $\tilde{f}(\mathbb{I})$*.*

Let $\mathrm{SUR}(\boldsymbol{\omega}) \overset{\Delta}{=} \max_{\mathbb{I} \in \mathcal{F}}\{\mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega}))\}$ denote the optimum value of the surrogate problem for trajectory profile $\boldsymbol{\omega}$. We now upper bound the optimum utility in the MARKOVIAN POI world (proved in full version). Our proof borrows ideas from the "prevailing reward argument" in [8].

**Lemma 1.** *For a* UTIL-MAX *problem with objective* $\mathsf{val}$ *and packing constraints* $\mathcal{F}$*, let* OPT *denote the utility of the optimal strategy. Then,*

$$\mathrm{OPT} \quad \leq \quad \mathbb{E}_{\boldsymbol{\omega}}[\mathrm{SUR}(\boldsymbol{\omega})] \quad = \quad \mathbb{E}_{\boldsymbol{\omega}}\big[\max_{\mathbb{I} \in \mathcal{F}}\{\mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega}))\}\big],$$

*where the expectation is over a random trajectory profile* $\boldsymbol{\omega}$ *that has every Markov system reaching a destination state.*

**Designing an Adaptive Strategy Using a Frugal Algorithm.** A FRUGAL algorithm selects elements one-by-one and irrevocably. Besides greedy algorithms, its definition also captures "non-greedy" algorithms such as primal-dual algorithms that do not have the reverse-deletion step [18].

**Definition 8 (FRUGAL Packing Algorithm).** *For a combinatorial optimization problem on universe* $J$ *in the* FREE-INFO *world with packing constraints* $\mathcal{F} \subseteq 2^J$ *and objective* $f : 2^J \to \mathbb{R}$*, we say Algorithm* $\mathcal{A}$ *is* FRUGAL *if there exists a* marginal-value *function* $g(\mathbf{Y}, i, y) : \mathbb{R}^J \times J \times \mathbb{R} \to \mathbb{R}$ *that is increasing in* $y$*, and for which the pseudocode is given by Algorithm 1. Note that this algorithm always returns a feasible solution if* $\emptyset \in \mathcal{F}$*.*

**Algorithm 1** FRUGAL Packing Algorithm $\mathcal{A}$

```
Start with $M = \emptyset$ and $v_i = 0$ for each element $i \in J$.
For each element $i \notin M$, compute $v_i = g(\mathbf{Y}_M, i, Y_i)$. Let $j = \arg\max_{i \notin M \text{ \& } M \cup i \in \mathcal{F}}\{v_i\}$.
If $v_j > 0$ then add $j$ into $M$ and go to Step 2. Otherwise, return $M$.
```

The following lemma shows that a FRUGAL algorithm can be converted to a strategy with the same utility in the MARKOVIAN POI world.

**Lemma 2.** *Given a* Frugal *packing Algorithm $\mathcal{A}$, there exists an adaptive strategy* $\mathrm{ALG}_{\mathcal{A}}$ *for the corresponding* Util-Max *problem in* Markovian PoI *world with utility at least* $\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))]$*, where* $\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})$ *is the solution returned by* $\mathcal{A}$ *for objective* $f(\mathbb{I}) = \mathsf{val}(\mathbf{Y}^{\max}(\boldsymbol{\omega}), \mathbb{I})$*.*

The strategy for Lemma 2 is in Algorithm 2 but the full proof is deferred.

**Algorithm 2** $\mathrm{ALG}_{\mathcal{A}}$ for Util-Max in Markovian PoI

```
1: Start with $M = \emptyset$ and $v_i = 0$ for all elements $i$.
2: For each element $i \notin M$, set $g(\mathbf{Y}^{\max}_M, i, \tau_i^{u_i})$ where $u_i$ is the current state of $i$.
3: Consider the element $j = \arg\max_{i \notin M \text{ \& } M \cup i \in \mathcal{F}} \{v_i\}$.
4: If $v_j > 0$, then if $\mathcal{S}_j$ is not in a destination state then proceed $\mathcal{S}_j$ by one step
   and go to Step 2. Else, when $v_j > 0$ but $\mathcal{S}_j$ is in a destination state $t_j$, select
   $j$ into $M$ and go to Step 2.
5: Else, if every element $i \notin M$ has $v_i \leq 0$ then return set $M$.
```

*Proof (Proof of Theorem 1).* From Lemma 2, the utility of $\mathrm{ALG}_{\mathcal{A}}$ is at least $\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))]$. Since Algorithm $\mathcal{A}$ is an $\alpha$-approx algorithm in the Free-Info world, it follows

$$\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))] \geq \frac{1}{\alpha} \cdot \mathbb{E}_{\boldsymbol{\omega}}\big[\max_{\mathbb{I} \in \mathcal{F}} \{\mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega}))\}\big].$$

Using the upper bound on optimal utility $\mathrm{OPT} \leq \mathbb{E}_{\boldsymbol{\omega}}\big[\max_{\mathbb{I} \in \mathcal{F}} \{\mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega}))\}\big]$ from Lemma 1, we have utility of $\mathrm{ALG}_{\mathcal{A}}$ is at least $\frac{1}{\alpha} \cdot \mathrm{OPT}$.

In the full version, a similar approach is used for the Disutil-Min problem with semi-additive function. This shows that for both Util-Max or Disutil-Min problem with semi-additive function, a Frugal algorithm can be transformed from Free-Info to Markovian PoI world while retaining its performance.

## 4 Robustness in Model Parameters

In practical applications, the parameters of Markov systems (i.e., transition probabilities, values, and prices) are not known exactly but are *estimated* by statistical sampling. In this setting, the *true parameters*, which govern how each Markov system evolves, differ from the estimated parameters that the algorithm uses to make decisions. This raises a natural question: how well does an adapted Frugal algorithm do when the true and the estimated parameters differ? We would hope to design a *robust* algorithm, meaning small estimation errors cause only small error in the utility objective.

In the important special case where the Markov chain corresponding to each element is formed by a *directed acyclic graph* (Dag), an adaptation of our strategy in Theorem 1 is robust. This Dag assumption turns out to be necessary as similar results do not hold for general Markov chains. In particular, we prove the following generalization of Theorem 1 under the Dag assumption.

**Theorem 2** (Informal statement)**.** *If there exists an $\alpha$-approximation* Frugal *algorithm $\mathcal{A}$ ($\alpha \geq 1$) for a packing problem with a semiadditive objective function, then it suffices to estimate the true model parameters of a* Dag-Markovian PoI *game within an additive error of $\epsilon/\mathrm{poly}$, where* poly *is some polynomial in the size of the input, to design a strategy with utility at least $\frac{1}{\alpha} \cdot \mathrm{OPT} - \epsilon$, where* OPT *is the utility of the optimal policy that knows all the* true *model parameters.*

Specifically, our strategy $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ for Theorem 2 is obtained from the strategy in Theorem 1 by making use of the following idea: each time we advance an element's Markov system, we slightly increase the estimated grade of every state in that Markov system. This ensures that whenever we advance a Markov system, we advance through an entire epoch and remain optimal in the "teasing game".

Our analysis of $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ works roughly as follows. We first show that under the Dag assumption, close estimates of the model parameters of a Markov system can be used to closely estimate the grade of each state. We can therefore assume that close estimates of all grades are given as input. Next we define the "shifted" prevailing cost corresponding to the "shifted" grades. This allows us to equate the utility of $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ by the utility of running $\mathcal{A}$ in the "modified" surrogate problem where the input to $\mathcal{A}$ is the "shifted" prevailing costs instead of the *true* prevailing costs. Finally, we prove that the "shifted" prevailing costs are close to the real prevailing costs and thus the "modified" surrogate problem is close to the surrogate problem. This allows us to bound the utility of running $\mathcal{A}$ in the "modified" surrogate problem by the optimal strategy to the surrogate problem. Combining with Lemma 1 finishes the proof of Theorem 2.

An analogous result for Disutil-Min also holds.

## 5 Handling Commitment Constraints

Consider the Markovian PoI model defined in §2 with an additional restriction that whenever we abandon advancing a Markov system, we need to *immediately* and *irrevocably* decide if we are selecting this element into the final solution $\mathbb{I}$. Since we only select ready elements, any element that is not ready when we abandon its Markov system is automatically discarded. We call this constraint *commitment.* The benchmark for our algorithm is the optimal policy *without* the commitment constraint. For single-stage probing, such commitment constraints have been well studied, especially in the context of stochastic matchings [2,4].

We study Util-Max in the Dag model with the commitment constraint. Our algorithms make use of the *online contention resolution schemes* (OCRSs) proposed in [10]. OCRSs address our problem in the Free-Info world[5] (i.e., we can see the realization of the r.v.s for free, but there is the commitment constraint). Constant factor "selectable" OCRSs are known for several constraint families: $\frac{1}{4}$ for matroids, $\frac{1}{2e}$ for matchings, and $\Omega(\frac{1}{k})$ for intersection of $k$ matroids [10]. We show how to adapt them to Markovian PoI with commitment.

[5] In fact, OCRSs consider a variant where the adversary chooses the order in which the elements are tried. This handles the present problem where we may choose the order.

**Theorem 3.** *For an additive objective, if there exists a $1/\alpha$-selectable OCRS ($\alpha \geq 1$) for a packing constraint $\mathcal{F}$, then there exists an $\alpha$-approximation algorithm for the corresponding* Dag-Util-Max *problem with commitment.*

The proof of this result uses a new LP relaxation (inspired from [13]) to bound the optimum utility of a Markovian PoI game *without* commitment (see §A.1). Although this relaxation is not exact even for Pandora's box (and cannot be used to design optimal strategies in Corollary 1), it turns out to suffice for our approximation guarantees. In §A.2, we use an OCRS to round this LP with only a small loss in the utility, while respecting the commitment constraint.

*Remark 1.* We do not consider Disutil-Min problem under commitment because it captures prophet inequalities in a minimization setting where no polynomial approximation is possible even for i.i.d. r.v.s [9, Theorem 4].

## 6 Related Work

Our work is related to work on multi-armed bandits in the scheduling literature. The Gittins index theorem [12] provides a simple optimal strategy for several scheduling problems where the objective is to maximize the long-term exponentially discounted reward. This theorem turned out to be fundamental and [19–21] gave alternate proofs. It can be also used to solve Weitzman's Pandora's box. The reader is referred to the book [11] for further discussions on this topic. Influenced by this literature, [8] studied scheduling of Markovian jobs, which is a minimization variant of the Gittins index theorem without any discounting. Their paper is part of the inspiration for our Markovian PoI model.

The Lagrangian variant of stochastic probing considered in [13] is similar to our Markovian PoI model. However, their approach using an LP relaxation to design a probing strategy is fundamentally different from our approach using a Frugal algorithm. E.g., unlike Corollary 1, their approach cannot give *optimal* probing strategies for matroid constraints due to an integrality gap. Also, their approach does not work for Disutil-Min. In Appendix A, we extend their techniques using OCRSs to handle the commitment constraint for Util-Max.

There is also a large body of work in related models where information has a price [1, 3, 5–7, 14–16]. Finally, as discussed in the introduction, the works in [17] and [18] are directly relevant to this paper. The former's primary focus is on *single item* settings and its applications to auction design, and the latter studies price of information in a *single-stage* probing model. Our contributions concern selecting *multiple items* in *multi-stage* probing model, in some sense unifying these two lines of work.

**Acknowledgements** A. Gupta and S. Singla were supported in part by NSF awards CCF1536002, CCF-1540541, and CCF-1617790, and the Indo-US Joint Center for Algorithms Under Uncertainty. H. Jiang was supported in part by CCF-1740551, CCF-1749609, and DMS-1839116. Z. Scully was supported by an ARCS Foundation scholarship and the NSF GRFP under Grant Nos. DGE-1745016 and DGE-125222.

# A Details for Handling Commitment Constraints

In this section we handle commitment constraints from §5 to prove Theorem 3. In §A.1, we give an LP relaxation to upper bound the optimum utility without the commitment constraint. In §A.2, we apply an OCRS to round the LP solution to obtain an adaptive policy, while satisfying the commitment constraint.

## A.1 Upper Bounding the Optimum Utility

Define the following variables, where $i$ is an index for the Markov systems.

- $y_i^u$: probability we reach state $u$ in Markov system $\mathcal{S}_i$ for $u \in V_i \setminus T_i$.
- $z_i^u$: probability we play $\mathcal{S}_i$ when it is in state $u$ for $u \in V_i \setminus T_i$.
- $x_i = \sum_{u \in T_i} z_i^u$: probability $\mathcal{S}_i$ is selected into the final solution when in a destination state.
- $P_{\mathcal{F}}$ is a convex relaxation containing all feasible solutions for packing $\mathcal{F}$.

We can now formulate the following LP, which is inspired from [13].

$$
\begin{array}{llr}
\max\limits_{\mathbf{z}} & \sum\limits_i \Big( \sum\limits_{u \in T_i} r_i^u z_i^u - \sum\limits_{u \in V_i \setminus T_i} \pi_i^u z_i^u \Big) & \\
\text{subject to} & y_i^{s_i} = 1 & \forall i \in J \\
& y_i^u = \sum_{v \in V_i} (P_i)_{uv} z_i^v & \forall i \in J, \forall u \in V_i \setminus s_i \\
& x_i = \sum_{u \in T_i} z_i^u & \forall i \in J \\
& z_i^u \leq y_i^u & \forall i \in J, \forall u \in V_i \\
& \mathbf{x} \in P_{\mathcal{F}} & \\
& x_i, y_i^u, z_i^u \geq 0 & \forall i \in J, \forall u \in V_i
\end{array}
$$

The first four constraints characterize the dynamics in advancing the Markov systems. The fifth constraint encodes the packing constraint $\mathcal{F}$. We denote the optimal solution of this LP as $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. We can efficiently solve the above LP for packing constraints such as matroids, matchings, and intersection of $k$ matroids.

If we interpret the variables $y_i^u, x_i$, and $z_i^u$ as the probabilities corresponding to the optimal strategy without commitment, it forms a feasible solution to the LP. This implies the following claim.

**Lemma 3.** *The optimum utility without commitment is at most the LP value.*

## A.2 Rounding the LP Using an OCRS

Before describing our rounding algorithm, we define an OCRS. Intuitively, it is an online algorithm that given a random set ground elements, selects a feasible subset of them. Moreover, if it can guarantee that every $i$ is selected w.p. at least $\frac{1}{\alpha} \cdot x_i$, it is called $\frac{1}{\alpha}$-selectable.

**Definition 9 (OCRS [10]).** *Given a point $x \in P_{\mathcal{F}}$, let $R(x)$ denote a random set containing each $i$ independently w.p. $x_i$. The elements $i$ reveal one-by-one whether $i \in R(x)$ and we need to decide irrevocably whether to select an $i \in R(x)$ into the final solution before the next element is revealed. An OCRS is an online algorithm that selects a subset $I \subseteq R(x)$ such that $I \in \mathcal{F}$.*

**Definition 10 ($\frac{1}{\alpha}$-Selectability [10]).** *Let $\alpha \geq 1$. An OCRS for $\mathcal{F}$ is $\frac{1}{\alpha}$-selectable if for any $x \in P_{\mathcal{F}}$ and all $i$, we have $\Pr[i \in I \mid i \in R(x)] \geq \frac{1}{\alpha}$.*

Our algorithm ALG uses OCRS as an oracle. It starts by fixing an arbitrary order $\pi$ of the Markov systems. (Our algorithm works even when an adversary decides the order of the Markov systems.) Then at each step, the algorithm considers the next element $i$ in $\pi$ and queries the OCRS whether to select element $i$ if it is ready. If OCRS decides to select $i$, then ALG advances the Markov system such that it plays from each state $u$ with independent probability $z_i^u/y_i^u$. This guarantees that the desination state is reached with probability $x_i$. If OCRS is not going to select $i$, then ALG moves on to the next element in $\pi$. A formal description of the algorithm can be found in Algorithm 3.

**Algorithm 3** Algorithm ALG for Handling the Commitment Constraint

```
1: Fix an arbitrary order $\pi$ of the items. Set $M = \emptyset$ and pass $\mathbf{x}$ to OCRS.
2: Consider the next element $i$ in the order of $\pi$. Query OCRS whether to add $i$ to $M$ if $i$ is ready.
   (a) If OCRS would add $i$ to $M$, then keep advancing the Markov system: play from each current state $u \in V_i \setminus T_i$ independently w.p. $z_i^u/y_i^u$, and otherwise go to Step 2. If a destination state $t$ is reached then add $i$ to $M$ w.p. $z_i^t/y_i^t$.
   (b) Go to Step 2.
```

We show below that ALG has a utility of at least $1/\alpha$ times the LP value.

**Lemma 4.** *The utility of* ALG *is at least $1/\alpha$ times the LP optimum.*

Since by Lemma 3 the LP optimum is an upper bound on the utility of any policy without commitment, this proves Theorem 3. We now prove Lemma 4.

*Proof (Proof of Lemma 4).* Recollect that we call a Markov system ready if it reaches an absorbing destination state. We first notice that once ALG starts to advance a Markov system $i$, then by Step 2 of Algorithm 3, element $i$ is ready with probability exactly $x_i$. This agrees with what ALG tells the OCRS. Since the OCRS is $1/\alpha$-selectable, the probability that any Markov system $\mathcal{S}_i$ begins advancing is $1/\alpha$. Here the probability is both over the random choice of the OCRS and the randomness due to the Markov systems. Conditioning on the event that $\mathcal{S}_i$ begins advancing, the probability that it is selected into the final solution on reaching a destination state $t \in T_i$ is exactly $z_i^t$. Hence, the conditioned utility from Markov system $\mathcal{S}_i$ is exactly

$$\textstyle\sum_{u \in T_i} r_i^u z_i^u - \sum_{u \in V_i \setminus T_i} \pi_i^u z_i^u. \tag{2}$$

By removing the conditioning and by linearity of expectation, the utility of ALG is at least $\frac{1}{\alpha} \cdot \sum_i \left( \sum_{u \in T_i} r_i^u z_i^u - \sum_{u \notin T_i} \pi_i^u z_i^u \right)$, which proves this lemma.

## References

1. Abbas, A.E., Howard, R.A.: Foundations of decision analysis. Pearson Higher Ed (2015)
2. Bansal, N., Gupta, A., Li, J., Mestre, J., Nagarajan, V., Rudra, A.: When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings. Algorithmica **63**(4), 733–762 (2012)
3. Charikar, M., Fagin, R., Guruswami, V., Kleinberg, J.M., Raghavan, P., Sahai, A.: Query strategies for priced information. J. Comput. Syst. Sci. **64**(4), 785–819 (2002). https://doi.org/10.1006/jcss.2002.1828, http://dx.doi.org/10.1006/jcss.2002.1828
4. Chen, N., Immorlica, N., Karlin, A.R., Mahdian, M., Rudra, A.: Approximating Matches Made in Heaven. In: ICALP (1). pp. 266–278 (2009)
5. Chen, Y., Immorlica, N., Lucier, B., Syrgkanis, V., Ziani, J.: Optimal data acquisition for statistical estimation. arXiv preprint arXiv:1711.01295 (2017)
6. Chen, Y., Hassani, S.H., Karbasi, A., Krause, A.: Sequential information maximization: When is greedy near-optimal? In: Conference on Learning Theory. pp. 338–363 (2015)
7. Chen, Y., Javdani, S., Karbasi, A., Bagnell, J.A., Srinivasa, S.S., Krause, A.: Submodular surrogates for value of information. In: AAAI. pp. 3511–3518 (2015)
8. Dumitriu, I., Tetali, P., Winkler, P.: On playing golf with two balls. SIAM Journal on Discrete Mathematics **16**(4), 604–615 (2003)
9. Esfandiari, H., Hajiaghayi, M., Liaghat, V., Monemizadeh, M.: Prophet secretary. SIAM Journal on Discrete Mathematics **31**(3), 1685–1701 (2017)
10. Feldman, M., Svensson, O., Zenklusen, R.: Online contention resolution schemes. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1014–1033. Society for Industrial and Applied Mathematics (2016)
11. Gittins, J., Glazebrook, K., Weber, R.: Multi-armed bandit allocation indices. John Wiley & Sons (2011)
12. Gittins, J., Jones, D.: A dynamic allocation index for the sequential design of experiments. Progress in statistics pp. 241–266 (1974)
13. Guha, S., Munagala, K.: Approximation algorithms for budgeted learning problems. In: STOC, pp. 104–113 (2007), full version as: *Approximation Algorithms for Bayesian Multi-Armed Bandit Problems*, http://arxiv.org/abs/1306.3525
14. Guha, S., Munagala, K., Sarkar, S.: Information acquisition and exploitation in multichannel wireless systems. In: IEEE Transactions on Information Theory. Citeseer (2007)
15. Gupta, A., Kumar, A.: Sorting and selection with structured costs. In: Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on. pp. 416–425. IEEE (2001)
16. Kannan, S., Khanna, S.: Selection with monotone comparison costs. In: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 10–17. Society for Industrial and Applied Mathematics (2003)
17. Kleinberg, R., Waggoner, B., Weyl, G.: Descending Price Optimally Coordinates Search. arXiv preprint arXiv:1603.07682 (2016)
18. Singla, S.: The price of information in combinatorial optimization. In: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM (2018)
19. Tsitsiklis, J.N.: A short proof of the Gittins index theorem. The Annals of Applied Probability pp. 194–199 (1994)

20. Weber, R.: On the Gittins index for multiarmed bandits. The Annals of Applied Probability **2**(4), 1024–1033 (1992)
21. Whittle, P.: Multi-armed bandits and the Gittins index. Journal of the Royal Statistical Society. Series B (Methodological) pp. 143–149 (1980)