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
[2] University of Washington, Seattle WA 98195, USA
[3] Princeton University, Princeton NJ 08544, USA

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. [33] and Singla [37], 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. [17] and Kleinberg et al. [33, Appendix G], who consider the multiple stage inspection aspect (ii) for the problem of selecting just a single site.

Our main results 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" α-approximation algorithms in the Free-Info setting into α-approximation adaptive algorithms in the Markovian PoI setting (Theorem 3.1). For example, this yields optimal algorithms for matroid optimization (Corollary 3.1).
- In §4 we show how to slightly modify our α-approximations for the Markovian PoI setting in Theorem 3.1 to make them robust to small changes in the model parameters (Theorem 4.1).
- In §5 we use *online contention resolution schemes* (OCRSs) [19] 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 α-approximation with commitment (Theorem 5.1).

The general idea behind our first result (Theorem 3.1) is the following. A Frugal combinatorial algorithm (Definition 3.6) 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 [17] (who did not consider values at the destinations).

Definition 2.1 (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 ω 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 2.2 (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 π_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 ω_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 [37], 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 3.1. *For a semiadditive objective function* val*, if there exists an α-approximation* Frugal *algorithm for a* Util-Max *problem over some packing constraints $\mathcal{F}$ in the* Free-Info *world, then there exists an α-approximation strategy for the corresponding* Util-Max *problem in the* Markovian PoI *world.*

We prove an analogous result for Disutil-Min in §D. The following corollaries immediately follow from known Frugal algorithms [37].

Corollary 3.1. *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* $\min\{f, \log n\}$*-approx for* Disutil-Min *for set-cover, where f 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 3.1, we define a *grade* for every state in a Markov system in §3.1, much as in [17]. 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 3.1. We consider Util-Max throughout, but analogous definitions and arguments hold for Disutil-Min.

3.1 Grade of a State

To define the *grade* τ^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 τ*-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 π^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 τ 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 τ, and one can also show that it is continuous [17, 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 τ from $-\infty$ to $+\infty$, the utility $U^v(\tau)$ becomes 0 at some critical value $\tau = \tau^v$. This critical value τ^v that depends on state v is the *grade.*

Definition 3.1 (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 τ_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 τ that makes the player *indifferent* between halting and playing. It is known how to compute grade efficiently [17, Section 7].

3.2 Prevailing Cost and Epoch

We now define a *prevailing cost* [17] and an *epoch.* The prevailing cost of Markov system $\mathcal{S}$ is its minimum grade at any point in time.

Definition 3.2 (Prevailing Cost). *The* prevailing cost *of Markov system $\mathcal{S}_i$ in a trajectory ω_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 ω 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 3.3 (Epoch). *An* epoch *for a trajectory ω is any maximal continuous segment of ω 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 3.1 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 3.4 (Semiadditive Function [37]). *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 3.1 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 3.1, 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 3.5 (Surrogate Problem). *Given a* UTIL-MAX *problem with semi-additive objective* 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. Our proof borrows ideas from the "prevailing reward argument" in [17].

Lemma 3.1. *For a* UTIL-MAX *problem with objective* 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.*

We prove Lemma 3.1 in §A.

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 [37].

Definition 3.6 (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 $\mathcal{A}$

```
1: Start with $M = \emptyset$ and $v_i = 0$ for each element $i \in J$.
2: For each element $i \notin M$, compute $v_i = g(\mathbf{Y}_M, i, Y_i)$. Let $j = \arg\max_{i \notin M \ \& \ M \cup i \in \mathcal{F}}\{v_i\}$.
3: 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 3.2. *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* $f(\mathbb{I}) = \mathsf{val}(\mathbf{Y}^{\max}(\boldsymbol{\omega}), \mathbb{I})$*.*

We prove Lemma 3.2 in §B. Finally, we can prove Theorem 3.1.

Proof (Proof of Theorem 3.1). From Lemma 3.2, 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 α-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 3.1, we have utility of $\mathrm{ALG}_{\mathcal{A}}$ is at least $\frac{1}{\alpha} \cdot \mathrm{OPT}$.

In §D, 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 3.1 is robust. This Dag assumption turns out to be necessary as similar results do not hold for general Markov chains (see Appendix F.1). In particular, we prove the following generalization of Theorem 3.1 under the Dag assumption.

Theorem 4.1 (Informal statement)**.** *If there exists an α-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 $\epsilon/$*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 4.1 is obtained from the strategy in Theorem 3.1 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 roughtly as follows. We first show that 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 3.1 finishes the proof of Theorem 4.1.

Similar arguments extend to prove the analogous result for DISUTIL-MIN.

We formally state our main theorem and the parameters on which it depends in Section 4.1. Section 4.2 shows that close estimates of transition probabilities can be used to obtain close estimates of the grades. In Section 4.3, we use these estimated grades to transform a FRUGAL algorithm into a robust adaptive algorithm for DAG-UTIL-MAX. Similar arguments can be used to obtain the corresponding results for DAG-DISUTIL-MIN (we omit this proof).

4.1 Main Results and Assumptions

We first explicitly define the *input size* of DAG-UTIL-MAX as follows.

(i) n is the number of Markov systems.
(ii) k is the maximum number of elements in a feasible solution, i.e., $k \triangleq \max_{\mathbb{I} \in \mathcal{F}} |\mathbb{I}|$.
(iii) D is the maximum depth of any DAG Markov system.

Denote B an upper bound on all input prices and values, i.e., $\forall i, \forall \pi \in \boldsymbol{\pi}_i, \forall r \in \mathbf{r}_i$, we have $|\pi| \leq B, |r| \leq B$. We make the following assumption.

Assumption 4.2 *The upper bound B is polynomial in n, k, and D.*

Such an assumption turns out to be necessary (see Appendix F.2). We now state our main theorem of this section.

Theorem 4.3. *Consider a* DAG-UTIL-MAX *problem with a semiadditive objective and satisfying Assumption 4.2. Suppose there exists an α-approximation* FRUGAL *algorithm in the* FREE-INFO *world. If each input parameter is known to within an additive error of ϵ/poly, where* poly *is some polynomial in n, k, and D, then there exists an adaptive algorithm $\widehat{\mathrm{ALG}}$ with utility at least*

$$\frac{1}{\alpha} \cdot \mathrm{OPT} - \epsilon,$$

where OPT *is the utility of the optimal policy that exactly knows the true input parameters.*

To simplify the proof of Theorem 4.3, we also assume the following without loss of generality (see Appendix F.3 for justifications).

(iv) All non-zero transition probabilities are lower bounded by $1/P$, where P is a polynomial in n, k, and D.

(v) We know the prices $\boldsymbol{\pi}$ and the rewards $\mathbf{r}$ exactly, i.e., the only unknown input parameters are the transition probabilities.

4.2 Well-Estimated Input Parameters Imply Well-Estimated Grades

We call the set of Markov systems constructed using our estimated transition probabilities the *estimated world.* The ith Markov system in this *estimated world* is denoted by $\widehat{\mathcal{S}}_i = (V_i, \widehat{P}_i, s_i, T_i, \boldsymbol{\pi}_i, \mathbf{r}_i)$, where $\widehat{P}_i$ contains the estimated transition probabilities. Note, $\boldsymbol{\pi}_i$ and $\mathbf{r}_i$ are exact due to Assumption (v). We estimate the grade of a state by simply computing the grade of that state in the estimated world. The following Lemma 4.1 bounds the error in estimated grades in terms of the error in transition probabilities.

Lemma 4.1. *Consider the* Dag-Util-Max *problem satisfying the assumptions in Section 4.1. Suppose all transition probabilities are estimated to within an additive error of $\epsilon < 1/P$, then $\forall i, \forall u \in V_i$, the estimated grade $\widehat{\tau}_i^u$ is within an additive factor of $O(L \cdot \epsilon)$ from the real grade τ_i^u, where $L = D^2BP$.*

Proof. We show below that $\tau_i^u \geq \widehat{\tau}_i^u - L \cdot \epsilon$. A symmetrical argument shows $\widehat{\tau}_i^u \geq \tau_i^u - L \cdot \epsilon$, which finishes the proof of this lemma.

We consider the Markov game $\widehat{G}_u$ defined in Section 3.1 in the estimated world. By definition, there exists an optimal policy Pol that advances the chain at least one more step and achieves an expected utility of 0. Also consider the Markov game G_u in the real world and apply Pol in G_u. Notice Pol might be sub-optimal in G_u and might therefore obtain a negative expected value. Let τ_{fair} be the cost τ in G_u such that Pol obtains an expected value of 0. It follows that $\tau_i^u \geq \tau_{fair}$. It therefore suffices to show that $\tau_{fair} \geq \widehat{\tau}_i^u - L \cdot \epsilon$.

Denote the set of trajectories when applying Pol (in either world) by $\mathcal{S}$ and those in which the item is picked by $\mathcal{S}_{win}$. Denote $p_{\boldsymbol{\omega}}$ the probability of a trajectory $\boldsymbol{\omega} \in \mathcal{S}$ in the real world and $\widehat{p}_{\boldsymbol{\omega}}$ the probability of it in the estimated world. Let $r_{\boldsymbol{\omega}}$ be the utility of $\boldsymbol{\omega}$ (as defined for Util-Max by ignoring the cost τ) in either world. It follows that

$$\tau_{fair} = \frac{1}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} p_{\boldsymbol{\omega}}} \cdot \sum_{\boldsymbol{\omega} \in \mathcal{S}} (p_{\boldsymbol{\omega}} \cdot r_{\boldsymbol{\omega}}) = \sum_{\boldsymbol{\omega} \in \mathcal{S}} \left(\frac{p_{\boldsymbol{\omega}}}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} p_{\boldsymbol{\omega}}} \cdot r_{\boldsymbol{\omega}} \right),$$

and that

$$\widehat{\tau}_i^u = \frac{1}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} \widehat{p}_{\boldsymbol{\omega}}} \cdot \sum_{\boldsymbol{\omega} \in \mathcal{S}} (\widehat{p}_{\boldsymbol{\omega}} \cdot r_{\boldsymbol{\omega}}) = \sum_{\boldsymbol{\omega} \in \mathcal{S}} \left(\frac{\widehat{p}_{\boldsymbol{\omega}}}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} \widehat{p}_{\boldsymbol{\omega}}} \cdot r_{\boldsymbol{\omega}} \right).$$

Since each transtion probability is lower bounded by $1/P$, it is estimated to within a multiplicative error of $(1 \pm O(P\epsilon))$. Since $p_{\boldsymbol{\omega}}$ and $\widehat{p}_{\boldsymbol{\omega}}$ can be written as the product of at most D probabilities, each term $\frac{p_{\boldsymbol{\omega}}}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} p_{\boldsymbol{\omega}}}$ is within a multiplicative error of $(1 \pm O(DP\epsilon))$ from $\frac{\widehat{p}_{\boldsymbol{\omega}}}{\sum_{\boldsymbol{\omega} \in \mathcal{S}_{win}} \widehat{p}_{\boldsymbol{\omega}}}$. It follows that τ_{fair} is within a multiplicative factor of $(1 \pm O(DP\epsilon))$ from $\widehat{\tau}_i^u$. But notice that $\widehat{\tau}_i^u \leq DB$, which implies that $\tau_{fair} \geq \widehat{\tau}_i^u - O(D^2BP \cdot \epsilon) = \widehat{\tau}_i^u - O(L \cdot \epsilon)$.

4.3 Designing an Adaptive Strategy for DAG-Utility Maximization

From the previous section we know how to obtain close estimates of the grades. Now we use well-estimated grades to design a robust adaptive strategy for Dag-Util-Max and prove Theorem 4.3. Theorem 4.3 directly follows by combining Lemma 3.1 and the following Lemma 4.2.

Lemma 4.2. *Assuming the conditions of Theorem 4.3 and that the grade of each state is estimated to within an additive factor of $\epsilon/4kD_i$, where D_i is the depth of $\mathcal{S}_i$, then there exists an adaptive algorithm $\widehat{\mathrm{ALG}}$ with utility at least*

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

To prove Lemma 4.2, we describe our algorithm $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ (Algorithm 2). We define $\widehat{\mathbf{Y}}^{\max}$ as follows.

Definition 4.1. *Fix a trajectory profile $\boldsymbol{\omega}$ where each Markov system reaches the destination state. For each i and $u \in V_i$, let $d_u(\omega_i)$ be the number of transitions for $\mathcal{S}_i$ to reach u from s_i by taking the trajectory $\omega_i \in \boldsymbol{\omega}$. Let $\widehat{\gamma}_i^u(\omega_i) = \widehat{\tau}_i^u + d_u(\omega_i)\epsilon/2kD_i$. Define $\widehat{Y}_{\omega_i}^{\max} \triangleq \min_{u \in \omega_i} \{\widehat{\gamma}_i^u(\omega_i)\}$. Denote the list of $\widehat{Y}_{\omega_i}^{\max}$'s as $\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega})$ and $\widehat{\mathbf{Y}}_M^{\max}(\boldsymbol{\omega})$ the list of $\widehat{Y}_{\omega_i}^{\max}$ values in the set M.*

The key idea in $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ (the main difference from Algorithm 4) is the "upward shifting" technique in Step 2. As we advance a Markov system, we shift our estimates of its grades upward. This guarantees that our algorithm is optimal in the teasing game G_T defined for Claim A.2.

Proof (Proof of Lemma 4.2). This lemma immediately follows from the following two claims (whose proofs are in Appendix E).

Claim. The utility of running $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ in the real world is exactly the same as

$$\mathbb{E}_{\boldsymbol{\omega}} \left[\mathsf{val}(Alg(\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega}), \mathcal{A}), \mathbf{Y}^{\max}(\boldsymbol{\omega})) \right].$$

Claim. For any trajectory profile $\boldsymbol{\omega}$ and for any i, $|\widehat{Y}_{\omega_i}^{\max} - Y_{\omega_i}^{\max}| \leq \epsilon/2k$. Thus

$$\mathsf{val}(Alg(\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega}), \mathcal{A}), \mathbf{Y}^{\max}(\boldsymbol{\omega})) \geq \frac{1}{\alpha} \cdot \max_{\mathbb{I} \in \mathcal{F}} \{ \mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega})) \} - \epsilon.$$

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

```
1: Start with $M = \emptyset$. Set $v_i = 0$ and $\mathrm{ctr}_i = 0$ for all elements $i$.
2: For each element $i \notin M$, set $v_i = g\left(\widehat{\mathbf{Y}}^{\max}_{M}, i, \widehat{\tau}^{u}_{i} + \mathrm{ctr}_i \cdot \epsilon/2kD_i\right)$ where $u$ is
   the current state of $i$.
3: Consider the element $j = \arg\max_{i \notin M \text{ \& } M \cup i \in \mathcal{F}}\{v_i\}$ and $v_j > 0$.
4: Proceed $\mathcal{S}_j$ for one step and set $\mathrm{ctr}_j = \mathrm{ctr}_j + 1$. If $t_j$ is reached by $\mathcal{S}_j$, select
   $j$ into $M$.
5: If every element $i \notin M$ has $v_i \leq 0$ then return set $M$. Else, go to Step 2.
```

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 [11,6].

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 [19]. OCRSs address our problem in the Free-Info world[4] (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 [19]. We show how to adapt them to Markovian PoI with commitment.

Theorem 5.1. *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 α-approximation algorithm for the corresponding* Dag-Util-Max *problem with commitment.*

The proof of this result uses a new LP relaxation (inspired from [22]) to bound the optimum utility of a Markovian PoI game *without* commitment (see §5.1). Although this relaxation is not exact even for Pandora's box (and cannot be used to design optimal strategies in Corollary 3.1), it turns out to suffice for our approximation guarantees. In §5.2, we use an OCRS to round this LP with only a small loss in the utility, while respecting the commitment constraint.

Remark 5.1. 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 [18, Theorem 4].

[4] 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.

In §5.1, we give an LP relaxation to upper bound the optimum utility without the commitment constraint. In §5.2, we apply an OCRS to round the LP solution to obtain an adaptive policy, while satisfying the commitment constraint.

5.1 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 [22].

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

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

5.2 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 5.1 (OCRS [19]). *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 5.2 ($\frac{1}{\alpha}$-Selectability [19]). *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 π 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 π 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 π. 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 5.2. *The utility of* ALG *is at least* $1/\alpha$ *times the LP optimum.*

Since by Lemma 5.1 the LP optimum is an upper bound on the utility of any policy without commitment, this proves Theorem 5.1. We now prove Lemma 5.2.

Proof (Proof of Lemma 5.2). 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

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

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.

6 Related Work

Our work is related to work on multi-armed bandits in the scheduling literature. The Gittins index theorem [21] 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 [38,39,41] gave alternate proofs. It can be also used to solve Weitzman's Pandora's box. The reader is referred to the book [20] for further discussions on this topic. Influenced by this literature, [17] 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 [22] 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 3.1, 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 §5, 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 [28,10,32,25,14,1,13,12]. Finally, as discussed in the introduction, the works in [33] and [37] 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.

The field of combinatorial optimization has been extensively studied: we refer the readers to Schrijver's popular book [36], and the references therein. In recent years, there has also been a lot of interest in studying these combinatorial problems for stochastic inputs. [15,16,24,22,9,34,35] considered stochastic knapsack, [11,2,6,8,3] studied stochastic matchings, [23,27,7] studied stochastic orienteering, [5,29,4,31,30] considered stochastic submodular maximization, and [22,23,26,35] studied budgeted multi-armed bandits. These works (besides [22]) do not consider mixed-sign utility objective or multi-stage probing, which is our primary focus.

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.

References

1. Abbas, A.E., Howard, R.A.: Foundations of decision analysis. Pearson Higher Ed (2015)

2. Adamczyk, M.: Improved analysis of the greedy algorithm for stochastic matching. Inf. Process. Lett. **111**(15), 731–737 (2011)
3. Adamczyk, M., Grandoni, F., Mukherjee, J.: Improved approximation algorithms for stochastic matching. In: Algorithms-ESA 2015, pp. 1–12. Springer (2015)
4. Adamczyk, M., Sviridenko, M., Ward, J.: Submodular stochastic probing on matroids. Mathematics of Operations Research **41**(3), 1022–1038 (2016)
5. Asadpour, A., Nazerzadeh, H., Saberi, A.: Stochastic submodular maximization. In: International Workshop on Internet and Network Economics. pp. 477–489. Springer (2008)
6. 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)
7. Bansal, N., Nagarajan, V.: On the adaptivity gap of stochastic orienteering. In: IPCO. pp. 114–125 (2014)
8. Baveja, A., Chavan, A., Nikiforov, A., Srinivasan, A., Xu, P.: Improved bounds in stochastic matching and optimization. In: APPROX. pp. 124–134 (2015)
9. Bhalgat, A., Goel, A., Khanna, S.: Improved approximation results for stochastic knapsack problems. In: SODA. pp. 1647–1665 (2011)
10. 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
11. Chen, N., Immorlica, N., Karlin, A.R., Mahdian, M., Rudra, A.: Approximating Matches Made in Heaven. In: ICALP (1). pp. 266–278 (2009)
12. Chen, Y., Immorlica, N., Lucier, B., Syrgkanis, V., Ziani, J.: Optimal data acquisition for statistical estimation. arXiv preprint arXiv:1711.01295 (2017)
13. 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)
14. 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)
15. Dean, B.C., Goemans, M.X., Vondrák, J.: Approximating the stochastic knapsack problem: The benefit of adaptivity. In: Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on. pp. 208–217. IEEE (2004)
16. Dean, B.C., Goemans, M.X., Vondrák, J.: Adaptivity and approximation for stochastic packing problems. In: SODA. pp. 395–404 (2005)
17. Dumitriu, I., Tetali, P., Winkler, P.: On playing golf with two balls. SIAM Journal on Discrete Mathematics **16**(4), 604–615 (2003)
18. Esfandiari, H., Hajiaghayi, M., Liaghat, V., Monemizadeh, M.: Prophet secretary. SIAM Journal on Discrete Mathematics **31**(3), 1685–1701 (2017)
19. 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)
20. Gittins, J., Glazebrook, K., Weber, R.: Multi-armed bandit allocation indices. John Wiley & Sons (2011)
21. Gittins, J., Jones, D.: A dynamic allocation index for the sequential design of experiments. Progress in statistics pp. 241–266 (1974)
22. 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
23. Guha, S., Munagala, K.: Multi-armed bandits with metric switching costs. In: ICALP. pp. 496–507 (2009)

24. Guha, S., Munagala, K.: Adaptive uncertainty resolution in bayesian combinatorial optimization problems. ACM Transactions on Algorithms (TALG) **8**(1), 1 (2012)
25. Guha, S., Munagala, K., Sarkar, S.: Information acquisition and exploitation in multichannel wireless systems. In: IEEE Transactions on Information Theory. Citeseer (2007)
26. Gupta, A., Krishnaswamy, R., Molinaro, M., Ravi, R.: Approximation algorithms for correlated knapsacks and non-martingale bandits. In: FOCS. pp. 827–836 (2011)
27. Gupta, A., Krishnaswamy, R., Nagarajan, V., Ravi, R.: Approximation algorithms for stochastic orienteering. In: SODA (2012), http://dl.acm.org/citation.cfm?id=2095116.2095237
28. 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)
29. Gupta, A., Nagarajan, V.: A stochastic probing problem with applications. In: IPCO. pp. 205–216 (2013)
30. Gupta, A., Nagarajan, V., Singla, S.: Algorithms and adaptivity gaps for stochastic probing. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1731–1747. SIAM (2016)
31. Gupta, A., Nagarajan, V., Singla, S.: Adaptivity Gaps for Stochastic Probing: Submodular and XOS Functions. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1688–1702. SIAM (2017)
32. 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)
33. Kleinberg, R., Waggoner, B., Weyl, G.: Descending Price Optimally Coordinates Search. arXiv preprint arXiv:1603.07682 (2016)
34. Li, J., Yuan, W.: Stochastic combinatorial optimization via poisson approximation. In: Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013. pp. 971–980 (2013). https://doi.org/10.1145/2488608.2488731, http://doi.acm.org/10.1145/2488608.2488731
35. Ma, W.: Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms: Extended abstract. In: SODA. pp. 1154–1163 (2014)
36. Schrijver, A.: Combinatorial optimization: polyhedra and efficiency, vol. 24. Springer Science & Business Media (2003)
37. Singla, S.: The price of information in combinatorial optimization. In: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM (2018)
38. Tsitsiklis, J.N.: A short proof of the Gittins index theorem. The Annals of Applied Probability pp. 194–199 (1994)
39. Weber, R.: On the Gittins index for multiarmed bandits. The Annals of Applied Probability **2**(4), 1024–1033 (1992)
40. Weitzman, M.L.: Optimal search for the best alternative. Econometrica: Journal of the Econometric Society pp. 641–654 (1979)
41. Whittle, P.: Multi-armed bandits and the Gittins index. Journal of the Royal Statistical Society. Series B (Methodological) pp. 143–149 (1980)

A Proof of Lemma 3.1

We restate Lemma 3.1 below.

Lemma 3.1. *For a* Util-Max *problem with objective* 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.*

Proof. We abuse the notation and use OPT to denote both the optimal policy and its utility. Suppose we fix a trajectory profile $\boldsymbol{\omega}$ where each Markov system $\mathcal{S}_i$ reaches a destination state. Let $\mathbb{I}(\boldsymbol{\omega})$ be the set of elements selected by OPT on $\boldsymbol{\omega}$, where notice that some of the unselected elements may not be ready: OPT might have selected $\mathbb{I}(\boldsymbol{\omega})$ only after playing prefixes of trajectories in $\boldsymbol{\omega}$. The following observation follows from the definition of $\mathrm{SUR}(\boldsymbol{\omega})$.

Observation A.1 *For any trajectory profile* $\boldsymbol{\omega}$*,*

$$\mathsf{val}(\mathbb{I}(\boldsymbol{\omega}), \mathbf{Y}^{\max}(\boldsymbol{\omega})) \leq \mathrm{SUR}(\boldsymbol{\omega}).$$

Now, using the following Lemma A.1 along with Observation A.1 finishes the proof of Lemma 3.1.

Lemma A.1. *The utility of the optimal strategy*

$$\mathrm{OPT} \leq \mathbb{E}_{\boldsymbol{\omega}}\left[\mathsf{val}(\mathbb{I}(\boldsymbol{\omega}), Y^{\max}(\boldsymbol{\omega}))\right].$$

Proof (Proof of Lemma A.1). Since for every trajectory profile $\boldsymbol{\omega}$ both OPT in the Markovian PoI world and $\mathbb{E}_{\boldsymbol{\omega}}\left[\mathsf{val}(\mathbb{I}(\boldsymbol{\omega}), Y^{\max}(\boldsymbol{\omega}))\right]$ in the Free-Info world pick the same set of elements $\mathbb{I}(\boldsymbol{\omega})$, the expected value due to the set function h is the same. Hence, WLOG assume $h(\mathbb{I}) = 0$ for all $\mathbb{I} \in \mathcal{F}$.

Now consider the following *teasing game* G_T defined using the prevailing cost from Definition 3.2. Consider a game where each Markov system $\mathcal{S}_i$ starts at its initial state s_i and a player is invited to advance the Markov systems. Besides advancing, the player is allowed to select any arbitrary elements (need not be feasible in $\mathcal{F}$) or terminate the game at any time during the game. Whenever an element i is selected, the player pays a corresponding *cost*, which is set to be the prevailing cost defined by the trajectory that lead to the current state in $\mathcal{S}_i$. The player's goal is to maximize the expected *value*, which is the expected utility (as defined for Util-Max) from advancing the Markov systems *minus* the expected total cost he pays when some items are selected. Observe that in this game the costs are updated in a "teasing" manner according to the prevailing costs that motivates the player to continue playing. By an argument similar to [17], we have the following lemma.

Lemma A.2. *The teasing game G_T is* fair*, which means that no strategy achieves a positive expected value by playing it and that there exists a strategy with zero expected value. Moreover, the following strategy plays fairly: irrespective of the order in which the Markov systems are played, whenever the player starts to advance a Markov system, he continues to advance it through the entire epoch.*

Now consider running the optimal policy OPT in the teasing game. Let $\boldsymbol{\omega}$ be a trajectory profile in which each chain reaches its destination state. Let $\boldsymbol{\omega}_T$ denote a trajectory profile until the moment when OPT returns the solution $\mathbb{I}(\boldsymbol{\omega})$ on the trajectory profile $\boldsymbol{\omega}$. It should be noticed that each trajectory in $\boldsymbol{\omega}_T$ is a prefix of the corresponding trajectory in $\boldsymbol{\omega}$. In particular, for an element $i \in \mathbb{I}(\boldsymbol{\omega})$, ω_i coincides with $(\boldsymbol{\omega}_T)_i$ since the destination state of $\mathcal{S}_i$ is reached. For an element $i \notin \mathbb{I}(\boldsymbol{\omega})$, however, $(\boldsymbol{\omega}_T)_i$ may only be a prefix of ω_i. It follows that applying OPT in G_T along trajectory profile $\boldsymbol{\omega}$ incurs a cost of $\sum_{i \in \mathbb{I}(\boldsymbol{\omega})} Y^{\max}_{(\boldsymbol{\omega}_T)_i}$, where $Y^{\max}_{(\boldsymbol{\omega}_T)_i}$ is the prevailing cost for $\mathcal{S}_i$ on trajectory $(\boldsymbol{\omega}_T)_i$ according to Definition 3.2. Since G_T is a fair game, the expected utility of OPT cannot be larger than the expected cost it pays, i.e.,

$$\mathrm{OPT} \le \mathbb{E}_{\boldsymbol{\omega}}\Big[\sum_{i \in \mathbb{I}(\boldsymbol{\omega})} Y^{\max}_{(\boldsymbol{\omega}_T)_i}\Big].$$

Since the elements $i \in \mathbb{I}(\boldsymbol{\omega})$ are ready, we have $\omega_i = (\boldsymbol{\omega}_T)_i$ and

$$\sum_{i \in \mathbb{I}(\boldsymbol{\omega})} Y^{\max}_{(\boldsymbol{\omega}_T)_i} = \sum_{i \in \mathbb{I}(\boldsymbol{\omega})} Y^{\max}_{\omega_i}.$$

This implies

$$\mathrm{OPT} \le \mathbb{E}_{\boldsymbol{\omega}}\Big[\sum_{i \in \mathbb{I}(\boldsymbol{\omega})} Y^{\max}_{\omega_i}\Big],$$

which finishes the proof of Lemma A.1.

B Proof of Lemma 3.2

We restate Lemma 3.2 below.

Lemma 3.2. *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* $f(\mathbb{I}) = \mathsf{val}(\mathbf{Y}^{\max}(\boldsymbol{\omega}), \mathbb{I})$*.*

Proof (Proof of Lemma 3.2). We describe how to adapt the Frugal Algorithm $\mathcal{A}$ to an adaptive strategy $\mathrm{ALG}_{\mathcal{A}}$ in the Markovian PoI world. $\mathrm{ALG}_{\mathcal{A}}$ uses the grade τ as proxy for $\mathbf{Y}^{\max}$, since $\mathbf{Y}^{\max}$ is known only when the Markov systems reach their destination states. More specifically, at each moment when the Frugal Algorithm $\mathcal{A}$ is trying to evaluate the marginal-value function for each element, instead of using the $\mathbf{Y}^{\max}$ value for each element, which we may not yet know at the moment, the strategy uses the τ values to compute the marginal. For the element chosen by $\mathcal{A}$, the corresponding Markov system will be advanced one more step. A more specific description of our algorithm $\text{ALG}_{\mathcal{A}}$ is given Algorithm 4. Here $\mathbf{Y}^{\max}_M$ for a set $M \subseteq J$ is defined as the list of $\mathbf{Y}^{\max}$ values that are in the set M.

Algorithm 4 $\text{ALG}_{\mathcal{A}}$ 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 $j = \arg\max_{i \notin M \ \&\ M \cup i \in \mathcal{F}} \{v_i\}$.
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 $M$.
```

In the following Claim B, we argue that for any trajectory profile $\boldsymbol{\omega}$, running $\text{ALG}_{\mathcal{A}}$ in Markovian PoI returns the same set of elements as running $\mathcal{A}$ for $\mathbf{Y}^{\max}(\boldsymbol{\omega})$.

Claim (Claim B). For any trajectory profile $\boldsymbol{\omega}$, the solution returned by running Algorithm 4 in the Markovian PoI world is the same as the solution by Algorithm $\mathcal{A}$ on $\mathbf{Y}^{\max}(\boldsymbol{\omega})$.

Before proving Claim B, we use it to prove Lemma 3.2 by showing that the utility of Algorithm 4 in the Markovian PoI world is at least

$$\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))].$$

By Claim B, the value due to the set function h is the same for both algorithms. So without loss of generality, assume h is always 0. We consider the teasing game G_T as defined in Claim A.2. By definition, g is an increasing function of the last parameter y. Since grade is used as that parameter and the grade of each state visited during an epoch is at least the grade of the initial state of that epoch, it follows that once Algorithm 4 starts to play a Markov system $\mathcal{S}_i$, it will not switch before finishing an epoch. Therefore, by Claim A.2, Algorithm 4 plays a fair game. So the expected cost that Algorithm 4 pays is the same as its expected utility from playing the Markov systems. However, Claim B gives the expected cost payed by Algorithm 4 is the same as the utility of running Algorithm $\mathcal{A}$ in the Free-Info world, i.e., $\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))]$. Hence, the utility of running Algorithm 4 is at least $\mathbb{E}_{\boldsymbol{\omega}}[\mathsf{val}(\mathcal{A}(\mathbf{Y}^{\max}(\boldsymbol{\omega})), \mathbf{Y}^{\max}(\boldsymbol{\omega}))]$.

It remains to prove the missing Claim B in the proof of Lemma 3.2.

Proof (Proof of Claim B). Suppose we fix a trajectory profile $\boldsymbol{\omega}$ where each Markov system reaches some destination state. We prove the claim by induction on the number of elements already selected into the set M. Suppose the set of elements selected into M is the same by running the two algorithms until now. We show that the next element selected by the algorithms into M is the same.

Assume for the purpose of contradiction that the next element picked by $\mathcal{A}$ is j but the next element picked by Algorithm 4 is $i \neq j$. By the definition of Algorithm $\mathcal{A}$,

$$j = \arg\max_{i' \notin M} \left\{ g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), i', Y_{\omega_{i'}}^{\max} \right) \right\}. \tag{2}$$

where ω_i' denotes the trajectory of $\mathcal{S}_{i'}$ in $\boldsymbol{\omega}$. Now we look at the trajectory ω_i, it follows that the prevailing cost $Y_{\omega_i}^{\max}$ is non-increasing over this trajectory and is equal to $Y_{\omega_i}^{\max}$ when $\mathcal{S}_i$ reaches the destination state. We look at the last moment t_0 when the prevailing cost of $\mathcal{S}_i$ decreases. Consider the first moment t_1 after t_0 that our Algorithm 4 decides to play $\mathcal{S}_i$ (but has not actually played $\mathcal{S}_i$ yet). It follows that the prevailing cost of $\mathcal{S}_i$ at moment t_1 is exactly the same as $Y_{\omega_i}^{\max}$ and also the grade $\tau_i^{u_i}$ of the current state u_i. Denote $Y_{\omega_j'}^{\max}$ the prevailing cost of $\mathcal{S}_j$ and u_j the state of $\mathcal{S}_j$ at moment t_1. Then we have $Y_{\omega_j'}^{\max} \geq Y_{\omega_j}^{\max}$ because the prevailing cost of $\mathcal{S}_j$ is also non-increasing. By the definition of t_1, one has

$$\begin{aligned} g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), i, Y_{\omega_i}^{\max}\right) &= g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), i, \tau_i^{u_i}\right) \\ &> g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), j, \tau_j^{u_j}\right) \geq g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), j, Y_{\omega_j'}^{\max}\right). \end{aligned}$$

However, since g is increasing in the last parameter, it follows that

$$g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), j, Y_{\omega_j'}^{\max}\right) \geq g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), j, Y_{\omega_j}^{\max}\right),$$

which implies

$$g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), i, Y_{\omega_i}^{\max}\right) > g\left(\mathbf{Y}_M^{\max}(\boldsymbol{\omega}), j, Y_{\omega_j}^{\max}\right).$$

This contradicts with the definition of j in Eq (2).

C Comparing Grade and Weitzman's Index for Pandora's Box

Recall Weitzman's Pandora's box formulation of the oil-drilling problem mentioned in Section 1. Given probability distributions of n independent random variables X_i (amount of oil at site i) and their *probing* (inspection) prices π_i, the goal is to design a strategy to *adaptively* probe a set Probed to maximize expected utility

$$\mathbb{E}\Big[\max_{i \in \mathsf{Probed}} \{X_i\} - \sum_{i \in \mathsf{Probed}} \pi_i\Big].$$

The Weitzman's index for site i, denoted by $\tau_i^{\max}$, is defined using the following equation $\mathbb{E}[(X_i - \tau_i^{\max})^+] = \pi_i$. It is known that the following strategy is optimal [40].

Selection Rule: The next site to be probed is the one with with the highest Weitzman's index.

Stopping Rule: Terminate when the maximum realized value amongst the probed sites exceeds the Weitzman's index of every unprobed site.

It turns out that Weitman's index $\tau_i^{\max}$ is simply the grade, defined in Section 3.1, in disguise. To see this, we start by noticing that each variable X_i with probing price π_i can be thought of as the following Markov system. There is one initial state s_i with moving cost π_i. s_i has transitions, with probabilities according to the distribution of X_i, to a set T_i of destination states, each corresponding to a possible outcome of the variable X_i . The value of each destination state is naturally set to be the corresponding outcome of X_i. We show below that $\tau_i^{\max}$ is simply the grade $\tau_i^{s_i}$ of the initial state s_i.

According to our definition of grade in Section 3.1, in the $\tau_i^{s_i}$-penalized Markov game $\mathcal{S}(\tau_i^{s_i})$, there is a fair strategy that probes site i and achieves a zero utility. Such a strategy would pick site i (i.e., play in the corresponding destination state) if and only if $X_i - \tau_i^{s_i} \geq 0$. The utility of that policy is thus $-\pi_i + \mathbb{E}[(X_i - \tau_i^{s_i})^+] = 0$. Comparing with the definition of Weitzman's index, this shows $\tau_i^{\max} = \tau_i^{s_i}$. The optimality of Weitzman's strategy is therefore also implied by Theorem 3.1.

D Adaptive Algorithms for Disutility Minimization

We give the corresponding definitions for the Disutil-Min problem.

Definition D.1 (Prevailing Reward for Disutil-Min). *The* prevailing reward *of $\mathcal{S}_i$ for the trajectory P_i in* Disutil-Min *is defined as*

$$R_{P_i}^{\min} \stackrel{\Delta}{=} \max_{u \in P_i} \{-\tau_i^u\}.$$

For a trajectory profile $\boldsymbol{\omega}$, denote $R_{\boldsymbol{\omega}}^{\min}$ the list of prevailing rewards for each Markov system.

For a trajectory P_i in the Disutil-Min problem, consider the change of the prevailing reward as the Markov system starts from s_i and moves according to P_i. It follows that the prevailing reward is non-decreasing in this process. Moreover, it increases whenever the Markov system reaches a state that has smaller grade than each previously visited state. Now we are ready to state the definition of an *epoch*.

Definition D.2 (Epoch for Disutil-Min). *An* epoch *is defined to be the period from the time when the prevailng reward increases until the moment just before the next time it increases.*

It follows that within an epoch, all states visited has grade no smaller than the prevailing reward at the start of this epoch and thus the prevailing reward stays constant in an epoch. We can therefore view the prevailing reward as a non-decreasing piece-wise constant function of time.

Definition D.3 (Frugal **Covering Algorithm).** *For a* Disutil-Min *problem in the* Deterministic *world with covering constraints $\mathcal{F}$ and cost function* cost*, 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 decreasing in y, and for which the pseudocode is given by Algorithm 5. Moreover, the function $g(\mathbf{Y}, i, y)$ should* encode *the constraints $\mathcal{F}$, such that whenever M is infeasible, then $\exists i \notin M$ with $v_i > 0$. This requirement will ensure that a feasible solution is returned.*

Algorithm 5 Frugal Covering Algorithm $\mathcal{A}$

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

With the definitions above, one can prove the following theorem for Disutil-Min using similar techniques as in Section 3.3.

Theorem D.1. *For a semiadditive objective function* cost*, if there exists an α-approximation* Frugal *algorithm for a* Disutil-Min *problem over some covering constraints $\mathcal{F}$ in the* Free-Info *world, then there exists an α-approximation strategy for the corresponding* Disutil-Min *problem in the* Markovian PoI *world.*

E Missing Proofs in the Robustness Model

Proof of Claim 4.3. Because $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ shifts the estimated grade upward by $\epsilon/2kD_i$ each time we advance $\mathcal{S}_i$ and that each grade is estimated to within an additive error of $\epsilon/2kD_i$, whenever $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ starts to advance a Markov system, it continues to advance it through the whole epoch. It follows from Claim A.2 that $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ is an optimal policy in the teasing game G_T. By a similar argument as the proof of Claim B, one can show that for any list of trajectories $\boldsymbol{\omega}$, running $\widehat{\mathrm{ALG}}_{\mathcal{A}}$ in the real world returns the same solution as running $\mathcal{A}$ on $\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega})$. These imply the claim. □

Proof of Claim 4.3. Since Markov system i can be played at most D_i times, it follows that the estimated grade is shifted upward by at most $(D_i - 1)\epsilon/2kD_i$. It follows that each estimated grade after the upward shifting is still within an additive error of $\epsilon/2k$ from the real grade, which finishes the first part of the grade.

The second part follows from the following inequalities.

$$\begin{aligned}
&\mathsf{val}(Alg(\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega}), \mathcal{A}), \mathbf{Y}^{\max}(\boldsymbol{\omega})) \\
&\geq \mathsf{val}(Alg(\widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega}), \mathcal{A}), \widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega})) - k \cdot \epsilon/2k \\
&\geq \frac{1}{\alpha} \cdot \max_{\mathbb{I} \in \mathcal{F}} \left\{ \mathsf{val}(\mathbb{I}, \widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega})) \right\} - \epsilon/2 \\
&\geq \frac{1}{\alpha} \cdot \mathsf{val} \left(\text{arg max}_{\mathbb{I} \in \mathcal{F}} \left\{ \mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega})) \right\}, \widehat{\mathbf{Y}}^{\max}(\boldsymbol{\omega}) \right) - \epsilon/2 \\
&\geq \frac{1}{\alpha} \cdot \max_{\mathbb{I} \in \mathcal{F}} \left\{ \mathsf{val}(\mathbb{I}, \mathbf{Y}^{\max}(\boldsymbol{\omega})) \right\} - \epsilon,
\end{aligned}$$

where the last line follows because $\alpha \geq 1$. □

F Assumptions in the Robustness Model

F.1 DAG Assumption

We give an example to illustrate why the DAG assumption is necessary for our robustness results to hold. We show that if there are cycles in the Markov chains, one might need to estimate the input parameters to a super-exponentially accurate precision in order to achieve a small additive loss in the performance.

Consider the following UTIL-MAX problem of picking at most one item (i.e. the constraint $\mathcal{F}$ is the uniform Matroid with rank 1) where *all the input parameters are polynomially bounded*. We have n Markov systems $\{\mathcal{S}_i\}_{1 \leq i \leq n}$. The last $n-2$ Markov systems each has only one state, which is a destination state, with value 0. These Markov systems can be safely ignored since one can pick nothing and obtains 0 utility. We can therefore focus only on the other two Markov systems.

The 2nd Markov system $\mathcal{S}_2$ has only one state, which is a destination state, with value 1. The first Markov system $\mathcal{S}_1$ has three states $\{s_1, v, t_1\}$, where s_1 is the initial state with playing cost $n^2/2^{2^n}$, t_i is the destination state with value $n^2/2$, and v is some intermediate state with playing cost 0. The transitions in $\mathcal{S}_1$ are as follows. s_1 goes to v deterministically. v goes to s_1 with probability $1 - 1/p2^{2^n}$ and t_1 with probability $1/p2^{2^n}$, where $p \in (0, 1]$. Notice that $\mathcal{S}_1$ contains a cycle and a negligible transition out of the cycle to the destination. It follows that the utility obtained by always playing $\mathcal{S}_1$ is $n^2/2 - pn^2$, which is $n^2/4$ if $p = 1/4$ and $-n^2/2$ if $p = 1$.

In this case, if we fail to estimate the transition probabilities of $\mathcal{S}_1$ to a super-exponentially accurate precision of $O(1/2^{2^n})$, it would render it impossible even to distinguish between the case where playing $\mathcal{S}_1$ has utility $\Theta(n^2)$ and the case where playing $\mathcal{S}_1$ has negative utility, which makes it impossible to obtain an approximation policy within a small additive error from the optimal policy.

F.2 Polynomial Upper Bound on Input Parameters

Here, we give an example to illustrate why Assumption 4.2 is necessary for our robustness results to hold. We show that if some parameters are exponential in the input parameter, then one might need to estimate some input parameters to within an additive error that is exponential in the input parameters.

Consider the following Util-Max problem of picking at most one item (i.e. the constraint $\mathcal{F}$ is the uniform Matroid with rank 1) where *all the input parameters are polynomially bounded*. We have n Markov systems $\{\mathcal{S}_i\}_{1\leq i\leq n}$. The last $n-1$ Markov systems deterministically give 0 utility. The first Markov system $\mathcal{S}_1$ has an initial state s_1 and two destination states t_1 and t_2. The initial state s_1 has price 3^n. It goes to t_1 with probability p and t_2 with probability $1-p$. t_1 has reward 2×3^n and t_2 has reward 0.

The player has to decide between playing $\mathcal{S}_1$ or doing nothing at all. If $p = 1/2+\Theta(1/2^n)$, then the utility of playing $\mathcal{S}_1$ is $\Theta(1.5^n)$ and if $p = 1/2-\Theta(1/2^n)$, then the utility of playing $\mathcal{S}_1$ is $-\Theta(1.5^n)$. It follows that one need to estimate the transition probabilities to within an additive error that is exponentially small.

F.3 Other Assumptions Without Loss of Generality

Recall that for the Dag-Util-Max problem in the robustness model, we made the following assumptions.

- All non-zero transition probabilities are lower bounded by $1/P$, where P is some polynomial in the parameters above.
- We can estimate the prices $\boldsymbol{\pi}$ and the rewards $\mathbf{r}$ exactly, i.e. the only unknown input parameters are the transition probabilities.

The assumption that all non-zero transition probabilities are polynomially lower bounded is without loss of generality. It can be removed by the following procedure. We start by setting a threshold $1/P$ and estimating all the data to within an additive error smaller than $1/P$. We then ignore the transitions that have estimated probabilities smaller than $2/P$. This is done by reallocating these probability masses to other transitions from the same state in both the original Markov systems and the estimated Markov systems. After the removal of these negligible transition probabilities, the remaining Markov systems have a lower bound of $1/P$ on all the transition probabilities. Since the maximum price paid on any sample path in a Markov system is at most DB, it follows that this changes the optimal policy by at most a very small additive factor if the polynomial P we take is large enough. Therefore, we shall assume without loss of generality a lower bound on all non-zero transition probabilities.

The assumption that we can estimate the prices $\boldsymbol{\pi}$ and the rewards $\mathbf{r}$ exactly is again without loss of generality and can be removed by the following argument with a small additive term in the theoretical guarantee. Suppose all the prices $\boldsymbol{\pi}$ and the rewards $\mathbf{r}$ are estimated within an additive error of δ/nD. Since one needs at most D steps to reach the destination for each Markov system, the utility is affected by at most a small additive factor of $\delta/nD\times nD=\delta$ if we set δ to be small. Therefore, we will assume that estimations of the prices $\boldsymbol{\pi}$ and the rewards $\mathbf{r}$ are exact and only the estimations of transition probabilities have deviations from the real transition probabilities.