Combinatorial Ski Rental and Online Bipartite Matching

HANRUI ZHANG and VINCENT CONITZER, Duke University

We consider a combinatorial variant of the classical ski rental problem — which we call combinatorial ski rental — where multiple resources are available to purchase and to rent, and are demanded online. Moreover, the costs of purchasing and renting are potentially combinatorial. The dual problem of combinatorial ski rental, which we call combinatorial online bipartite matching, generalizes the classical online bipartite matching problem into a form where constraints, induced by both offline and online vertices, can be combinatorial. We give a 2-competitive (resp. $e/(e-1)$-competitive) deterministic (resp. randomized) algorithm for combinatorial ski rental, and an $e/(e-1)$-competitive algorithm for combinatorial online bipartite matching. All these ratios are optimal given simple lower bounds inherited from the respective well-studied special cases. We also prove information-theoretic impossibility of constant-factor algorithms when any part of our assumptions is considerably relaxed.

CCS Concepts: • **Theory of computation → Online algorithms**; **Packing and covering problems**.

Additional Key Words and Phrases: ski rental, online matching, online primal-dual

ACM Reference Format:
Hanrui Zhang and Vincent Conitzer. 2020. Combinatorial Ski Rental and Online Bipartite Matching. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC '20), July 13–17, 2020, Virtual Event, Hungary.* ACM, New York, NY, USA, 32 pages. https://doi.org/10.1145/3391403.3399470

1 INTRODUCTION

Combinatorial ski rental. We started a company. At each time t, a job that needs to be done arrives. We may have existing employees that can perform the job, in which case there is no cost to us. If we do not, we may outsource the job at a one-time cost. More generally, our employees may be able to perform part of the job, in which case we have to outsource the remainder of the job. If S is the set of skills that our employees do *not* have, let $g_t(S)$ be the cost that we pay to outsource the remainder of the job that arrives at time t. (Note that this cost depends on t, so that, for example, a certain skill may not be needed at all at a given time t and therefore not contribute to the cost.) We assume g_t is submodular: the more skills we do not have available in house, the less it costs to cover an additional skill at the margin — for example because the same company to which we outsource has multiple of the needed skills and gives discounts.

Of course, our employees are not free either. Moreover, we operate in an environment where we cannot lay off employees. We think of hiring an employee as paying a large one-time cost after which the employee will work for free. (For example, we give the employee a one-time equity share, or we just think of the present value of all the employee's future salary.) Assume that every employee has exactly one skill. Let $f(S)$ be the cost of hiring a set of employees with skills S. We again assume that f is submodular. (A natural special case would be where f is linear, but it is possible that employees are excited to work with specific other employees and therefore willing to

EC '20, July 13–17, 2020, Virtual Event, Hungary

ACM ISBN 978-1-4503-7975-5/20/07…$15.00
https://doi.org/10.1145/3391403.3399470

be paid less if those employees are there.) Note that f is unchanging over time; the same types of employees are always available at the same costs.

The question is now at what point we make the (irreversible) decision to add an employee / skill to our company, without knowing which jobs show up in the future. Let $S_t \subseteq [n]$ be the set of skills that we have in the company at time t and let T be the final time at which we operate. Our goal is to minimize $f(S_T) + \int_0^T g_t([n] \setminus S_t)\,\mathrm{d}t$. This problem is a combinatorial generalization of the *ski rental* problem [29].

Combinatorial online bipartite matching. We run an organization that relies heavily on volunteer effort — say, a homeless shelter. We have a set of tasks $[n]$, known from the beginning, to accomplish (e.g., preparing certain meals, making beds). At each time t, a set of volunteers shows up to help us accomplish tasks; we can assign them to tasks. Let $\lambda_t(i)$ denote the number of volunteers that arrived at time t and were assigned to task i. However, there are constraints on how we assign the volunteers: for example, some volunteers may not be able to do certain tasks, or prefer not to work on only one task. For each subset S of tasks, let $g_t(S)$ denote the maximum number of volunteers that we can feasibly assign to this subset of tasks. Note that a fraction of a volunteer may be assigned to a task. We assume g_t is submodular: the more tasks that are already in the set, the fewer additional volunteers we can place by adding another task to the set. We also have overall constraints on how much of each task can be done: let $f(S)$ denote the maximum amount of work that can be usefully done on the set of tasks S. We assume f is submodular. (A natural special case would be where f is linear, but there may be interactions: for example, if we make more of one type of meal, that will reduce the number of another type of meal for which we would have a use.)

The question is now to which tasks to assign volunteers coming in, without knowing which volunteers will show up in the future. Our goal is to maximize the number of tasks done, $\int_0^T \sum_{i\in[n]} \lambda_t(i)\,\mathrm{d}t$.

Note that the problem could also be phrased in terms of assigning crowdworkers to tasks. Our problem is a generalization of the online bipartite matching problem [30]. That line of work is usually motivated in terms of determining which ads to show to users, when there is a budget for how often to show each ad and users appear over time. Our framework (as well as previous work [36]) allows for the budget to be for combinations of ads, where $f(S)$ is the combined budget. Unlike previous work, our framework also allows for there to be complex constraints on which combination of ads to show to a user, where $g_t(S)$ is the maximum number of ads from set S that can be shown to the user.

1.1 Our Results

Our results are roughly twofold:

- We study combinatorial ski rental and combinatorial online bipartite matching in the online primal-dual framework. Based on the observation that the LP formulations of the two problems are the dual problem of each other, we construct a primal-dual update rule, which maintains feasible primal and dual solutions over time, and keeps the primal / dual objective values within an $e/(e-1)$ factor of each other. As a result, we give
  - a deterministic (resp. randomized) 2-competitive (resp. $e/(e-1)$-competitive) algorithm for the combinatorial ski rental problem, and
  - an $e/(e-1)$-competitive algorithm for the combinatorial online bipartite matching problem.

  All these ratios are tight given simple lower bounds inherited from well-studied special cases of the respective problems.

- We then argue that for combinatorial ski rental, the assumptions we make (and particularly the class of cost functions we consider) are in fact necessary for a constant competitive ratio. We give information-theoretic lower bounds, ruling out the possibility of
  - $o(\sqrt{\log n})$-competitive algorithms when the cost of purchasing is allowed to be XOS,
  - $o(\log n)$-competitive algorithms when the cost of renting is allowed to be supermodular, or
  - $O(n^{1-\varepsilon})$-competitive algorithms for any $\varepsilon > 0$ when *upgrading* is not allowed. (By "upgrading" being allowed, we mean that the total cost accrued from purchases depends only on the final set of purchased resources, and not on the order in which they are purchased.)

To the best of our knowledge, we are the first to consider the ski rental problem with *fully combinatorial costs*, whether offline or online. Going beyond the assumption that resources are independent of each other, our model captures the possibility of complex interactions among resources, which, as the examples illustrate, are ubiquitous in practice. The most general model considered prior to our work [36] allows the offline cost, i.e., the cost of purchasing, to be combinatorial, but requires the online cost, i.e., the cost of renting, to be a mild generalization of additive costs, to the extent that such costs still admit succinct representations. Our main algorithmic contribution here can be summarized as follows:

> ***Even when there is complex interaction between resources, the combinatorial ski rental problem admits exactly the same (optimal) competitive ratios as the well-studied special case of a single resource.***

This does require that we formulate the combinatorial ski rental problem such that (1) the costs are submodular (as opposed to general set functions), and (2) the cost of purchasing is for "upgrading," as opposed to separate purchase. One may naturally wonder whether these conditions are necessary. We answer this question by proving a number of lower bounds, which can be summarized as follows:

> ***For the combinatorial ski rental problem, constant competitive ratios cannot be achieved in environments where upgrading is not allowed, or where resources exhibit complementarity, or even slight non-submodularity.***

Our results give a relatively complete picture for the combinatorial ski rental problem.

As for online bipartite matching, our formulation replaces edges with combinatorial capacity constraints, which can equivalently be viewed as polymatroid feasibility constraints. As illustrated in the example, our formulation of the problem models the possibility of complex constraints induced by both offline vertices and online vertices. Our contribution here can be summarized as follows:

> ***Even in the presence of two-sided combinatorial capacity constraints, the combinatorial online bipartite matching problem admits the same $(1 - 1/e)$ competitive ratio as the classical online bipartite matching problem.***

Technically, en route to our algorithmic and hardness results, we develop, among other techniques, (1) a systematic approach for combinatorial covering-packing problems in the online primal-dual framework, and (2) a symmetrization technique tailored for lower bounds involving XOS functions, which may be of independent interest.

1.2 Technical Overview

To construct our algorithms for combinatorial ski rental and online bipartite matching, we adopt the celebrated online primal-dual framework, and generalize the standard primal-dual analysis of the single-resource ski rental problem to combinatorial domains. Henceforth, we refer to combinatorial ski rental as the primal problem, and combinatorial online bipartite matching as the dual problem.

Our goal is to update primal and dual variables over time, such that (1) all constraints are satisfied at all times, and (2) the ratio between the primal and dual objectives is bounded by a constant. By weak duality, this constant gives us precisely our desired competitive ratio for both problems, which we set to be $e/(e-1)$. En route, we face the following key technical difficulties.

- There are exponentially many primal variables and exponentially many dual constraints. Our first step is to enforce strong structure on the family of update rules that we consider, so that the primal variables are always completely determined by the dual variables. As a result, when designing our update rule, we only need to consider the n dual variables.
- The primal objective and some dual constraints are *global*, i.e., they involve integration over time. These components of the LPs appear more challenging to deal with under the online primal-dual framework. To handle these components, we show that they can be reduced to the local components of the LPs, by imposing additional local conditions on the dual variables. This allows us to effectively decouple the time-dependent aspect, and focus on the snapshot of the problem at each time.
- Now the problem reduces to establishing the existence of a group of update rates, satisfying all local constraints at any time. To this end, we observe that the update procedure can be interpreted as racing among the n dual variables. We prove that at any time, the dual variables can be divided into groups according to their current location and speed, and moreover, the update rule can be designed separately for each group. Within each group, we show that the existence of feasible rates of updating is equivalent to the existence of a solution to a carefully constructed LP which achieves objective value at least 1. We prove the existence of the latter by considering the dual of the LP, i.e., we analyze the structure of any locally minimal dual solution, and show any such solution must induce objective value at least 1.

As for the lower bounds, we heavily exploit Yao's Minimax Lemma, and construct hard distributions for all three relaxations of the combinatorial ski rental problem. One particularly interesting technical ingredient is a symmetrization argument for XOS functions. For the relaxation where the purchasing cost can be XOS, we first give an idealized construction relying on the assumption that we can hide future clauses, which is not possible in our model. We remove this assumption via a symmetrization argument, which works by, roughly speaking, preparing all possible clauses from the very beginning, where different realizations of the clause in the same phase impose costs on different sets of resources. We then effectively add clauses by demanding resources appearing in the right realization of the clause we wish to add. The above trick symmetrizes the purchasing cost f, and implements the idealized construction.

1.3 Related Work

Most closely related to our results is the highly insightful work by Wang and Wong [36]. They consider important special cases of both the primal and the dual problems that we consider, namely, the matroid online vertex cover problem[1] and the matroid online bipartite matching problem. The key difference between our results and theirs is that they do not consider online combinatorial costs / constraints. Technically, the absence of such costs and constraints substantially simplifies the problem, and in particular, makes the task of designing efficient investment strategies, which is one of our key technical contributions, almost trivial.

[1] In the online vertex cover problem, there are some offline vertices, and other vertices arrive online. When an online vertex arrives, a decision must be made whether to include it or not. In the latter case, all of its offline neighbors must be included. This is the special case of our combinatorial ski rental problem where the offline vertices are the resources/skills, f is additive, and g_t is the weight of the corresponding online vertex if at least one of the needed resources for that task has not yet been purchased, and 0 otherwise. The matroid version of the problem allows the cost of including offline vertices to be combinatorial, which is still a special case of our model.

Online optimization with combinatorial constraints or objectives has been studied in various contexts. We name a few results related to ours. Hazan and Kale [23] consider submodular minimization in the no-regret learning context, where the goal is to achieve sublinear regret. Kapralov et al. [27] consider submodular welfare maximization when items arrive online, and show that greedy gives the optimal competitive ratio of 2. Devanur et al. [15] consider submodular welfare maximization when bidders arrive online. Chan et al. [13] study an online generalization of submodular maximization, which they call submodular online bipartite matching with matroid constraints and free disposal. These results are incomparable with ours, in the sense that they modify or generalize the submodular minimization / maximization problem in orthogonal directions.

Since the seminal work by Karlin et al. [29], numerous applications of the ski rental problem (e.g., TCP acknowledgement) have been found [17, 28]. Variants of the problem have also been considered. Madry and Panigrahi [32] consider the ski rental problem in a stochastic setting. Lotker et al. [31] generalize the original problem to allow multiple options interpolating between renting and buying to be available. Gollapudi and Panigrahi [22] consider a variant of the problem where expert advice is available. These results are incomparable with ours, since none of them consider combinatorial costs.

Following the groundbreaking work by Karp et al. [30], variants of the online bipartite matching problem have also been studied. Examples include the vertex-weighted version [1, 16], adwords [10, 14, 21, 34], and display ads allocation [19, 20]. Recently, the problem has been generalized to the two-sided [35] and fully online models [24, 26], where in both models all vertices may arrive online, and in the latter model, the graph is not necessarily bipartite. For more on online matching and advertisement, see the survey by Mehta [33].

The online primal-dual framework [10, 16] has proved useful in various problems, including online set cover [2], paging [8, 9], k-server [7], network design [3, 4], routing [11], load balancing and machine scheduling [5, 25], and matching [16]. The framework has also been generalized to handle convex costs [6]. See the survey by Buchbinder and Naor [12] for more on the online primal-dual framework.

2 PRELIMINARIES

Throughout the paper, we use n to denote the number of resources in combinatorial ski rental, as well as the number of offline vertices in combinatorial online bipartite matching. We use $[n] = \{1, 2, \ldots, n\}$ to denote the set of positive integers not exceeding n. For a set function $f : 2^{[n]} \to \mathbb{R}_+ = [0, \infty)$ and two sets $S, T \subseteq [n]$, let $f(S \mid T) = f(S \cup T) - f(T)$ be the marginal value of S given T.

2.1 Classes of Set Functions

In this paper, we consider the following classes of set functions.

- *Additive functions.* A set function $f : 2^{[n]} \to \mathbb{R}_+$ is additive if for any $S, T \subseteq [n]$ where $S \cap T = \emptyset$, $f(S) + f(T) = f(S \cup T)$.
- *Submodular functions.* A set function $f : 2^{[n]} \to \mathbb{R}_+$ is submodular if for any $S \subseteq [n]$ and $T' \subseteq T \subseteq [n]$, $f(S \mid T) \leq f(S \mid T')$.
- *XOS functions.* A set function $f : 2^{[n]} \to \mathbb{R}_+$ is XOS if there exists $k \in \mathbb{N}$ and k additive functions $c_1, \ldots, c_k : 2^{[n]} \to \mathbb{R}_+$ over the same domain, such that for any $S \subseteq [n]$, $f(S) = \max_{j \in [k]}\{c_j(S)\}$. We call $\{c_j\}_{j \in [k]}$ the *clauses* of f.
- *Supermodular functions.* A set function $f : 2^{[n]} \to \mathbb{R}_+$ is supermodular, if for any $S \subseteq [n]$ and $T' \subseteq T \subseteq [n]$, $f(S \mid T) \geq f(S \mid T')$.

It is known that all additive functions are submodular, and all submodular functions are XOS (see, e.g., [18]). Intuitively, additive functions model values or costs of independent resources, submodular functions capture the notion of diminishing marginal return / cost, and XOS functions are considered "just beyond" submodular functions. Supermodular functions, in contrast, model resources as complements to each other, and they are generally considered difficult to deal with. We prove our upper bounds when all cost functions are submodular, and consider XOS and supermodular functions in our lower bounds.

2.2 The Combinatorial Ski Rental Problem

We consider a continuous formulation of the problem, which not only generalizes discrete time models, but is also allows for cleaner algorithms and analyses. Let $N = [n]$ be the set of resources, which can be purchased or rented, and are demanded online. Let $f : 2^{[n]} \to \mathbb{R}_+$ denote the cost of purchasing, and $g : \mathbb{R}_+ \times 2^{[n]} \to \mathbb{R}_+$ the time-dependent unit-time cost of renting. The cost of renting S from time t_1 to t_2 is

$$\int_{t_1}^{t_2} g(t, S)\,\mathrm{d}t.$$

For simplicity, we assume $g(\cdot, S)$ is piecewise constant for any S. Also we require f and g_t for any $t \geq 0$ to be submodular, normalized (i.e., $f(\emptyset) = g_t(\emptyset) = 0$) and monotone. For any $t \geq 0$, let $g_t(\cdot) = g(t, \cdot)$. The set of resources demanded at time t is implicitly encoded in g_t — if a resource i is not demanded, then for all $S \subseteq [n]$, $g_t(\{i\} \mid S) = 0$. Equivalently, one may think of this as all resources are demanded at any time, but for some items, the marginal renting cost is always 0, i.e., they are available for free.

The actions of the algorithm are reflected in a time-dependent set of purchased resources, S_t. Before time 0 the set of purchased resources is $S_{<0} = \emptyset$. At each time $t \geq 0$, suppose the set of resources already purchased before t is $S_{<t} = \bigcup_{t'<t} S_{t'}$. The algorithm observes g_t (and remembers $g_{t'}$ for any $t' < t$), and must choose a (possibly empty) set ΔS_t to purchase. Formally, we require S_t to be a function of f, $\{g_{t'} \mid t' \leq t\}$ and the random bits if the algorithm is randomized, and for any $t_1 < t_2$, $S_{t_1} \subseteq S_{t_2}$. The cost of purchasing ΔS_t is $f(\Delta S_t \mid S_{<t})$, and the new set of purchased resources is $S_t = S_{<t} \cup \Delta S_t$. We note that the cost paid is for "upgrading" from $S_{<t}$ to S_t, rather than separate purchase — the problem with separate purchase is strictly harder, and as we show, admits no constant factor algorithm. For the set of unpurchased resources, $[n] \setminus S_t$, the algorithm incurs cost $g_t([n] \setminus S_t)\,\mathrm{d}t$ for renting those resources. This continues until time $T \geq 0$, which is unobservable by the algorithm. At time T, all resources stop to be demanded, and the total cost of the algorithm can then be calculated as

$$\mathsf{ALG}(f, g, T) = f(S_T) + \int_0^T g_t([n] \setminus S_t)\,\mathrm{d}t.$$

The goal is to minimize this total cost. Note that the integral is well-defined, since S_t can change (expand, in fact) at most n times. Also, the order of purchasing and renting at some particular time does not matter — we could replace the renting cost at time t by $g_t([n] \setminus S_{<t})\,\mathrm{d}t$. The total cost remains the same since $g_t([n] \setminus S_t)$ and $g_t([n] \setminus S_{<t})$ as functions of t are identical almost everywhere.

2.3 The Combinatorial Online Bipartite Matching Problem

Again, we consider a continuous formulation. Let $N = [n]$ be the set of offline vertices, which can be (fractionally) matched to online "vertices" upon their arrival. Let $f : 2^{[n]} \to \mathbb{R}_+$ denote the global capacity constraint over the offline vertices, and $g_t : 2^{[n]} \to \mathbb{R}_+$ the local capacity constraint for the online vertex arriving at time t. Intuitively, f describes how many online vertices overall can be matched to each subset of the offline vertices, and g_t describes how much of the online vertex / supply arriving at time t can be matched to each subset of the offline vertices. Again, we require f and g_t to be submodular, normalized, and monotone.

The actions of the algorithm are reflected in the assignment of the online supply to the n offline vertices. Let $\lambda_t(i) \geq 0$ be the amount of supply at time t assigned to offline vertex i. The assignment must satisfy the global and local capacity constraints, i.e., for any $t \in \mathbb{R}_+$ and $S \subseteq [n]$,

$$\int_0^t \sum_{i \in S} \lambda_\tau(i)\, \mathrm{d}\tau \leq f(S) \quad \text{and} \quad \sum_{i \in S} \lambda_t(i) \leq g_t(S).$$

Again, the online environment enforces that λ_t can only depend on f and $\{g_\tau\}_{\tau \leq t}$. At time T, online supply / vertices stop to arrive, and the total amount of supply matched can be calculated as

$$\mathsf{ALG}(f, g, T) = \int_0^T \sum_{i \in [n]} \lambda_t(i)\, \mathrm{d}t.$$

The goal is to maximize this total amount of supply matched.

2.4 The Offline Optimum and the Competitive Ratio

We define the competitive ratio for the combinatorial ski rental problem — the respective definitions for the combinatorial online bipartite matching problem is totally similar.

Given f, T, and g_t for $0 \leq t \leq T$, the offline optimal cost is given by

$$\mathsf{OPT}(f, g, T) = \min_S \left\{ f(S) + \int_0^T g_t([n] \setminus S)\, \mathrm{d}t \right\}.$$

The competitive ratio of an algorithm is then defined to be

$$\sup_{f,g,T} \left\{ \frac{\mathsf{ALG}(f, g, T)}{\mathsf{OPT}(f, g, T)} \right\},$$

where the sup is taken over all submodular cost functions.

3 WARMUP: THE SINGLE-RESOURCE CASE

In this section, we review the primal-dual approach to the classical single-resource ski rental problem, which provides important intuition for our results. The construction and ideas therein are also present in previous work based on the online primal-dual framework (see, e.g., [12]). We present and interpret them here in order to provide intuition for our algorithm and analysis in the fully combinatorial case.

Notation. We consider the case where $n = 1$. Let $f = f(\{1\})$ be the cost of purchasing the only resource, and $g_t = g_t(\{1\})$ be the time-dependent rate for renting the resource. One could w.l.o.g. assume $g_t = 1$. However, here we allow g_t to vary to provide more intuition for the general case, where no similar simplifying assumptions can be made.

The primal-dual formulation. Consider the following standard online LP formulation and its dual. Although the formulation takes a differential form, it is easy to check that weak duality still holds. The primal formulation is clearly a relaxation of the ski rental problem.

Algorithms via primal-dual update. We now show how to construct a deterministic 2-competitive algorithm and a randomized $e/(e-1)$-competitive algorithm utilizing the above formulation. The plan is to maintain feasible primal and dual solutions, while ensuring that the cumulative primal cost is at most twice or $e/(e-1)$ times the dual cost at any time. In particular, we increase variable x

$$
\begin{array}{llll}
\textbf{Primal:} & \min x \cdot f + \int_0^T y_t \cdot g_t \, \mathrm{d}t & & \textbf{Dual:} \quad \max \int_0^T \lambda_t \, \mathrm{d}t \\
& \text{s.t. } x + y_t \geq 1 & \forall t \in [0, T] & \text{s.t. } \int_0^T \lambda_t \, \mathrm{d}t \leq f \\
& x, y_t \geq 0 & \forall t \in [0, T] & \lambda_t \leq g_t \qquad \forall t \in [0, T]
\end{array}
$$

Fig. 1. The online primal-dual formulation in the single-resource case.

as the cost of renting g_t arrives online. To this end, we make variable x time-dependent, and denote its value at time t by x_t. Observe that at any moment t, for optimality we may set $y_t = 1 - x_t$. This ensures the primal solution to be feasible and minimizes the primal cost. Also, for reasons that will be clear momentarily, we set $\lambda_t = g_t$ whenever $x_t < 1$, and $\lambda_t = 0$ otherwise — this is our dual update rule for both of our algorithms. The rule automatically satisfies the second dual constraint.

We first give the deterministic primal update rule, where $x_t \in \{0, 1\}$. The rule is simple, and its corresponding algorithm is extremely well-known: set $x_t = 1$ as soon as the cumulative renting cost $\int_0^t g_\tau \, \mathrm{d}\tau$ exceeds f. In order to establish the competitive ratio of 2, we need to show: (1) the first dual constraint is satisfied, and (2) the primal cost is upper bounded by twice the dual cost at any time t. For (1), observe that for any t, $\int_0^t \lambda_\tau \, \mathrm{d}\tau \leq \int_0^t g_\tau \, \mathrm{d}\tau$, and at the moment when the cumulative renting cost reaches f, x_t is set to 1 and no further update will be made to any of the variables. Therefore the first dual constraint is satisfied. For the ratio of 2, observe that the primal cost is

$$x_t \cdot f + \int_0^t (1 - x_\tau) g_\tau \, \mathrm{d}\tau = x_t \cdot f + \int_0^t (1 - x_\tau) \lambda_\tau \, \mathrm{d}\tau.$$

When the cumulative renting cost is less than f, $x_t = 0$, and the primal cost is precisely the dual cost. Otherwise, $x_t = 1$, the dual cost is f and the primal cost is

$$1 \cdot f + \int_0^t 1 \cdot \lambda_\tau \, \mathrm{d}\tau = 2f.$$

The ratio follows.

Now we turn to the randomized algorithm, where x_t can take any value in $[0, 1]$. Suppose x_t is differentiable. At time t, the derivative of the primal cost is

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(x_t \cdot f + \int_0^t y_\tau \cdot g_\tau \, \mathrm{d}\tau \right) = \frac{\mathrm{d}}{\mathrm{d}t} x_t \cdot f + y_t \cdot g_t = \frac{\mathrm{d}}{\mathrm{d}t} x_t \cdot f + (1 - x_t) \cdot g_t.$$

On the other hand, the derivative of the dual cost is simply λ_t. For reasons that will be clear momentarily, we set $\lambda_t = g_t$ whenever $x_t < 1$. To guarantee the desired ratio of $e/(e-1)$, we want the primal derivative to be exactly $e/(e-1)$ times the dual derivative, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t} x_t \cdot f + (1 - x_t) \cdot g_t = \frac{e}{e-1} g_t \Longrightarrow g_t x_t + \frac{1}{e-1} g_t = f \frac{\mathrm{d}}{\mathrm{d}t} x_t.$$

Given boundary condition $x_0 = 0$, the above equation is solved by

$$x_t = \frac{1}{e-1} \left(\exp \left(\int_0^t g_\tau \, \mathrm{d}\tau / f \right) - 1 \right).$$

In light of this, we replace g_τ in the above equation with λ_τ, and set

$$x_t = \frac{1}{e-1}\left(\exp\left(\int_0^t \lambda_\tau \,\mathrm{d}\tau/f\right) - 1\right).$$

This is because whenever $x_t < 1$, $\lambda_t = g_t$, and when $x_t = 1$, there is no need to update any of the variables. It remains to show primal and dual feasibility at any moment t. The primal side is clearly feasible. For the dual, we need to show that at time t, if $\int_0^t g_\tau \,\mathrm{d}\tau \geq f$, then $\lambda_t = 0$, or equivalently, $x_t = 1$. This is again trivial, because assuming otherwise, i.e., $x_t < 1$, we have

$$x_t = \frac{1}{e-1}\left(\exp\left(\int_0^t \lambda_\tau \,\mathrm{d}\tau/f\right) - 1\right) = \frac{1}{e-1}\left(\exp\left(\int_0^t g_\tau \,\mathrm{d}\tau/f\right) - 1\right) \geq \frac{1}{e-1}\left(\exp(1) - 1)\right) = 1,$$

a contradiction.

We remark that x_t in the above solution is indeed monotonically non-decreasing. A direct corollary is that the above primal solution can be rounded online, preserving the cost in expectation. The rounding scheme is simple: pick r from $[0, 1]$ uniformly at random, maintain the primal and dual variables, and purchase the resource as soon as $x_t \geq r$.

Take-away messages. The most important insight to be gained from the above analysis is the formula for x_t in the randomized algorithm and its components. We now try to interpret this. First we isolate a key component of the formula. Let $q_t = \frac{1}{f}\int_0^t \lambda_\tau \,\mathrm{d}\tau$. Observe that there is a one-to-one correspondence between x_t and q_t, i.e., $x_t = \frac{1}{e-1}(\exp(q_t) - 1)$. Moreover, x_t is monotone in q_t, $x_t = 0$ iff $q_t = 0$, and $x_t = 1$ iff $q_t = 1$. We call q_t the *investment* into the resource, which directly determines to what extent the resource is available. For our purposes, it is much more convenient to work with q_t than with x_t.

Now we interpret the cost rate of renting, g_t. In the above construction, before the resource is completely purchased, g_t contributes its entire value to the dual cost (i.e., $g_t = \lambda_t$), which up to a factor of $e/(e-1)$ upper bounds the primal cost. One may view this as a *budget* of amount $g_t \,\mathrm{d}t$ at time t, which can be invested into the resource immediately. The amount actually invested is $\lambda_t \,\mathrm{d}t$. Moreover, by considering q_t instead of x_t, investing has a *linear* interpretation: the unit cost of investing in the resource is f; by investing $\lambda_t \,\mathrm{d}t$ at time t, one gains $\lambda_t \,\mathrm{d}t/f$ shares of the resource. This investment interpretation lies in the core of our algorithm in the combinatorial case.

With the investment interpretation in mind, we review the two algorithms. In both our algorithms, the investment rule is essentially the same — invest immediately all incoming budget into the resource, as long as the capacity of the resource allows. The only difference between the two scenarios and the two ratios is the mapping from the investment q_t to the actual fraction of the resource owned, x_t. In the deterministic case, the mapping takes 1 to 1, and anything smaller than 1 to 0. The extreme non-smoothness of this mapping enforces the competitive ratio to be as large as 2. Intuitively, the investment pays back only upon maturity. As a result, the algorithm has completely no access to the resource before maturity, and suffers from the entirety of the primal cost. In the randomized case, however, the mapping is much more smooth, which in some sense resembles continuous compounded interest. The investment then immediately turns into fractional ownership of the resource, which can be used to reduce the renting cost on the fly, and allows for a better competitive ratio.

4 HANDLING COMBINATORIAL OBJECTIVES AND CONSTRAINTS

In this section, we present and analyze our algorithm for combinatorial ski rental and online bipartite matching (Algorithm 1), unified under the online primal-dual framework. In Section 4.1, we give the LP formulations that we consider, and interpret its components. In Section 4.2, we elaborate on the

Algorithm 1: the primal-dual update algorithm.

```
Input  : the purchasing cost (or the offline capacity constraint) $f$ and the renting cost (or
         the online budget constraint) $g_t$.
Output : the primal distribution of purchased set of resources $\{x_t(S)\}_S$, and the dual
         investment rates (or fractions of online vertex matched) $\{\lambda_t(i)\}_i$ at any time $t$.
At time $t = 0$, initialize $q_t(i) = 0$ for each resource $i$.
for any time $t \geq 0$ do
    Compute the order over the resources $\prec_t = \prec(f, g_t, q_t)$, as defined in Section 4.4.
    Let $\{\lambda_t(i)\}_i$ be the group of investment rates defined in Lemma 4.7.
    for each resource $i$ where $q_t(i) < 1$ do
        Let the unit price for resource $i$ be $h_t(i) = f(S(\sim_t i) \mid S(\prec_t i)) \cdot \lambda_t(i)/\lambda_t(S(\sim_t i))$, as
          defined in Section 4.2.
        Update $q_t(i)$ such that $\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = \lambda_t(i)/h_t(i)$.
        if deterministic then
            Let $p_t(i) = \mathbb{I}[q_t(i) = 1]$.
        end
        else
            Let $p_t(i) = \frac{1}{e-1}(\exp(q_t(i)) - 1)$.
        end
    end
    Let $\{x_t(S)\}_S$ be such that $x_t(S) = \Pr_{\theta \sim \mathcal{U}(0,1)}[S = \{i \mid p_t(i) \geq \theta\}]$, as discussed in
      Section 4.2.
end
```

market interpretation of the LP formulation, and focus our attention to a structured class of update rules, which relies in a blackbox manner on a order over resources to be defined later. In Section 4.3, we define a class of update rules (or investment strategies given the market interpretation), namely efficient investment strategies, and reduce the problem of designing competitive algorithms for both our problems to the problem of designing efficient investment strategies. In Section 4.4, we define and motivate the order used in our update rule, and prove its properties. In Section 4.5, we construct an update rule, and show that it satisfies all requirements of efficient strategies, and therefore implies competitive online solutions to the primal and dual LPs. Finally in Section 4.6, we show how to round the primal solution online to give a randomized algorithm for the combinatorial ski rental problem.

4.1 The Online Primal-dual Formulation

We start from a combinatorial point of view, and consider the following online primal-dual formulation of both problems that we consider (Figure 2). Below we interpret the variables and constraints. In the primal formulation, which corresponds to the combinatorial ski rental problem, $x(S)$ is the probability that set S is purchased, or the fraction of S owned. One may alternatively view $x(S)$ as a distribution over the set purchased, which is possibly empty. Although not explicitly required, we always ensure $\sum_S x(S) = 1$ in our update rule. Similarly, $y_t(S)$ is the fraction of set S to be rented at time t. The primal objective is simply the total fractional (or expected) cost of purchasing and renting resources. Recall that we define the problem in such a way, that all resources are demanded at all times. The first primal constraint says each resource must be available (in expectation) at each time t, being owned, rented, or (fractionally) both. In the dual formulation, which corresponds to the combinatorial online bipartite matching problem, $\lambda_t(i)$ is the fraction of online vertex at time t matched to the i-th offline vertex. To understand our update rule and its analysis, it is sometimes more convenient to view $\lambda_t(i)$ as the rate of investing into resource i. The objective is to maximize the total amount matched (or the total investment, which can be interpreted as providing as much budget for the primal as possible). The second dual constraint is a local budget constraint, controling the possible investment rates into all resources at each time. The first dual constraint, on the other hand, is a global capacity constraint on the offline vertices (or resources), which says one cannot over-invest into any set of resources.

$$\begin{array}{llll}
\textbf{Primal:} & \min & \displaystyle\sum_{S\subseteq[n]} x(S)\cdot f(S) + \int_0^T \sum_{S\subseteq[n]} y_t(S)\cdot g_t(S)\,\mathrm{d}t & \\
& \text{s.t.} & \displaystyle\sum_{S\subseteq[n]:i\in S} x(S) + y_t(S) \geq 1 & \forall i\in[n], t\in[0,T] \\
& & x(S), y_t(S) \geq 0 & \forall S\subseteq[n], t\in[0,T]
\end{array}$$

$$\begin{array}{llll}
\textbf{Dual:} & \max & \displaystyle\int_0^T \sum_{i\in[n]} \lambda_t(i)\,\mathrm{d}t & \\
& \text{s.t.} & \displaystyle\int_0^T \sum_{i\in S} \lambda_t(i)\,\mathrm{d}t \leq f(S) & \forall S\subseteq[n] \\
& & \displaystyle\sum_{i\in S} \lambda_t(i) \leq g_t(S) & \forall S\subseteq[n], t\in[0,T]
\end{array}$$

Fig. 2. The online primal-dual formulation in the general case.

4.2 Understanding the Market

The combinatorial setting, while generalizing the single-resource case, is fundamentally more complex than the latter in several ways. We discuss relevant aspects of the complexity here, which help simplify the problem and better motivate the investment interpretation. Throughout our discussion of the update rule, $x(S)$ is again time-dependent. We use $x_t(S)$ to denote its value at time t, as in the single-resource analysis.

Individual investments. First observe that despite the costs being combinatorial, the dual implicitly allows / requires investing in individual resources, rather than combinations of them. In light of this observation, we also separate individual resources in the primal formulation. Instead of $x_t(S)$, we consider $p_t(i) \in [0, 1]$ for each resource i, at any time t. $\{p_t(i)\}_i$ determine $\{x_t(S)\}_S$ in the following way. For any set S,

$$x_t(S) = \Pr_{\theta\sim\mathcal{U}(0,1)}[S = \{i \mid p_t(i) \geq \theta\}],$$

where $\mathcal{U}(0, 1)$ is the uniform distribution over $[0, 1]$. In words, $x_t(S)$ is the probability that S is the set of resources i whose $p_t(i)$ exceeds a uniformly random threshold θ from $[0, 1]$. $p_t(i)$ is then the probability that resource i appears in such a random set. Clearly $\sum_S x_t(S) = 1$ at any time t. Given $p_t(i)$, for optimality we always set $y_t(S)$ in the following way throughout this section. Let

$y_t(S) = x_t([n] \setminus S)$. Or equivalently, for any $S \subseteq [n]$,

$$y_t(S) = \Pr_{\theta \sim \mathcal{U}(0,1)}[S = \{i \mid p_t(i) < \theta\}].$$

$p_t(i)$ here in some sense plays the role of x_t in the single-resource formulation. So, similarly we define the investment $q_t(i)$ in resource i, which directly determines $p_t(i)$. The relation between $p_t(i)$ and $q_t(i)$ are the same as in the single-resource setting, i.e., $p_t(i) = \mathbb{I}[q_t(i) = 1]$ in the deterministic case, and $p_t(i) = \frac{1}{e-1}(\exp(q_t(i)) - 1)$ in the randomized case. As in the single-resource setting, it is much easier to work with the investments $q_t(i)$ rather than the probabilities $p_t(i)$.

Unit prices in the combinatorial case. Now we consider the unit prices of investing in the resources. The prices can be derived by opening up the analysis which we present later. Here we give only the prices, and omit the reasoning behind them, which will be clear momentarily. In the combinatorial case, the unit prices of investing generally are no longer fixed, but may depend on what resources one already controls and the rates at which one is spending the budget. At any time t, we define a weak order (where equality is allowed) $\prec_t = \prec(f, q_t, \lambda_t)$ over the resources, which is determined by the cost of purchasing f, the investments q_t and the investment rates λ_t. For two resources i, j, if $q_t(i) < q_t(j)$ then $i \prec_t j$. When $q_t(i) = q_t(j)$, we need more delicate conditions involving f and λ_t to determine the order between i and j. In such cases, it is possible that $i \prec_t j$, $j \prec_t i$, or $i \not\prec_t j$ and $j \not\prec_t i$, i.e., i is "equal" to j according to $\prec_t$, denoted by $i \sim_t j$. Given $\prec_t$, let $S(\succ_t i)$ be the suffix defined by i, i.e., $S(\succ_t i) = \{j \mid j \succ_t i\}$. Similarly we define $S(\prec_t i) = \{j \mid j \prec_t i\}$. Let $S(\sim_t i)$ be the set of resources equal to i, including i itself, i.e., $S(\sim_t i) = \{j \mid j \sim_t i\}$. Given $\prec_t$, $S(\succ_t i)$ and $S(\sim_t i)$, the unit price of investing in i is then defined to be

$$h_t(i) = f(S(\sim_t i) \mid S(\succ_t i)) \cdot \frac{\lambda_t(i)}{\lambda_t(S(\sim_t i))} = f(S(\sim_t i) \mid S(\succ_t i)) \cdot \frac{\lambda_t(i)}{\sum_{j \in S(\sim_t i)} \lambda_t(j)}.$$

Here, we abuse notation to allow $\lambda_t(S) = \sum_{i \in S} \lambda_t(i)$. When $S(\sim_t i) = \{i\}$, $h_t(i)$ simplifies to $h_t(i) = f(\{i\} \mid S(\succ_t i))$.

Note that in certain cases, it is possible that $h_t(i) = 0$. This happens when (1) $f(S(\sim_t i) \mid S(\succ_t i)) = 0$, or (2) $\lambda_t(i) = 0$. To avoid the first case, we assume $f(i \mid [n] \setminus i) > 0$ for any i, by, for example, replacing f with f' where $f'(S) = f(S) + \varepsilon \cdot |S|$. This is without loss of generality, since letting $\varepsilon \to 0$, f' approximates f up to any precision. When the second case happens, $\lambda_t(i) = h_t(i) = 0$, and we define

$$\frac{\lambda_t(i)}{h_t(i)} = \frac{\lambda_t(S(\sim_t i))}{f(S(\sim_t i) \mid S(\succ_t i))}.$$

We formally define $\prec$ later — details of $\prec$ are immaterial for our current discussion. Here, the only properties of $\prec$ we need are as follows.

Lemma 4.1 (stability almost everywhere). *When λ_t is piecewise constant in t, $\prec_t$ changes only on a zero-measure subset of $[0, T]$. As a result, for any i, $h_t(i)$ is constant in t almost everywhere.*

As we shall see, λ_t being piecewise constant is not a trivial property as it appears to be. Our construction of λ_t, as we show, is in fact piecewise constant given that g_t is piecewise constant.

Lemma 4.2 (monotone rates). *Fix f, some resource i, and two groups of investments and rates (q_t, λ_t) and (q'_t, λ'_t). Let $S(\succ_t i) = \{j \mid q_t(j) > q_t(i)\}$ and $S(\succ'_t i) = \{j \mid q'_t(j) > q'_t(i)\}$, and similarly $S(\succeq_t i) = \{j \mid q_t(j) \geq q_t(i)\}$ and $S(\succeq'_t i) = \{j \mid q'_t(j) \geq q'_t(i)\}$. Suppose (q_t, λ_t) and (q'_t, λ'_t) satisfy $\lambda_t(j) \geq \lambda'_t(j)$ for all j and $S(\succ_t i) \supseteq S(\succ'_t i)$ and $S(\succeq_t i) \supseteq S(\succeq'_t i)$. Let $\prec_t = \prec(f, q_t, \lambda_t)$, and $\prec'_t = \prec(f, q'_t, \lambda'_t)$. Let $h_t(i)$ and $h'_t(i)$ be the unit prices for i induced by $\prec_t$ and $\prec'_t$ respectively. Then $\frac{\lambda_t(i)}{h_t(i)} \geq \frac{\lambda'_t(i)}{h'_t(i)}$.*

Lemma 4.2 crucially depends on f being submodular. We will prove these lemmas when we formally define $\prec$.

To see why the unit prices are appropriately defined, suppose all resources have different investments, i.e., for any $i \neq j, q_t(i) \neq q_t(j)$. Recall that the fractional / expected cost of the purchased set of resources is $\sum_S x_t(S) \cdot f(S)$, where $x_t(S)$ are determined by $p_t(i)$ in the way described above. Consider investing in some resource i. For sufficiently small ε, when we increase the probability of $i, p_t(i)$, by ε, we are intuitively upgrading an ε fraction of the set $S(\succ_t i)$ to $S(\succ_t i) \cup \{i\}$. That is, we decrease $x_t(S(\succ_t i))$ by ε and increase $x_t(S(\succ_t i) \cup \{i\})$ by ε, simultaneously. As a result, we pay a marginal cost of

$$\varepsilon \cdot f(\{i\} \cup S(\succ_t i)) - \varepsilon \cdot f(S(\succ_t i)) = \varepsilon \cdot f(\{i\} \mid S(\succ_t i)) = \varepsilon \cdot h_t(i).$$

The effective unit price of resource i here is exactly $h_t(i)$.

Dynamics of the market. At time t, given the rates of investing λ_t, the cost of purchasing f and the investments q_t together define how the market evolves (i.e., the update rule), explicitly given by

$$\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = \frac{\lambda_t(i)}{h_t(i)}.$$

We consider only update rules given by the above equation. Note that for $i \sim_t j$, $\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = \frac{\mathrm{d}}{\mathrm{d}t} q_t(j)$. That is, the investments into incomparable resources always increase at the same rate. In certain pathological cases, the above dynamics may be ill-defined. We therefore assume any regularity conditions on λ_t (and g_t, which in our construction partly determines λ_t) for the dynamics to make sense. For simplicity, we consider these quantities to be piecewise constant functions of time.

Our task is to choose the investment rates λ_t subject to the budget and capacity constraints, and show the induced update rule gives our desired competitive ratios. We break this into two parts. Within this environment, we first (1) set our investment goals, which as we show, would imply the desired competitive ratios, and then (2) present our investment strategy, i.e., construction of λ_t, to achieve these goals. The next subsections are dedicated to these two tasks.

4.3 Setting Investment Goals

As in the single-resource case, we construct our algorithm by updating the primal and dual solutions, keeping both of them feasible and within some factor of each other. In the above subsection, we formulate this as an investment problem by enforcing certain structures of the update rule and setting individual but time-dependent unit prices for all resources. Yet, it is not clear what our objectives might be in this environment. The single-resource analysis suggests that one should probably invest as much as possible at all times. However, in the combinatorial case, complex correlation (encoded as constraints involving multiple resources) may prevent us from investing in a single-minded fashion — we must set reasonable goals and build our portfolio carefully to meet them. We now describe our investment goals, and show how they imply the desired competitive ratios. The investment strategy used to achieve the goals is postponed to the next subsections.

For notational simplicity, for any $\theta \in [0, 1]$, let $S(<_t \theta) = \{i \mid q_t(i) < \theta\}$. Similarly, let $S(>_t \theta) = \{i \mid q_t(i) > \theta\}$ and $S(=_t \theta) = \{i \mid q_t(i) = \theta\}$. Consider the following requirements.

- *Piecewise constant:* $\forall i \in [n]$, $\lambda_t(i)$ is piecewise constant in t.
- *Budget feasibility:* $\forall t \in \mathbb{R}_+, S \subseteq [n]$, $\lambda_t(S) = \sum_{i \in S} \lambda_t(i) \leq g_t(S)$. This is the second dual constraint, which says the investment strategy must abide by the budget constraint at all times.

- *Group-wise full spending:* $\forall t \in \mathbb{R}_+, \theta \in [0, 1)$,
$$\lambda_t(S(=_t \theta)) = \sum_{i \in S(=_t \theta)} \lambda_t(i) = g_t(S(=_t \theta) \mid S(<_t \theta)).$$
This requirement means for each group of resources partially owned to a same extent, we must invest all incoming budget available in this group into resources in the same group.
- *No wasting:* $\forall t \in \mathbb{R}_+, i \in [n]$, $q_t(i) = 1 \implies \lambda_t(i) = 0$. In other words, we never invest in a fully owned resource. This implies in particular that $q_t(i)$ is always no larger than 1.

If an investment strategy satisfies these requirements, we say the strategy is *efficient*. One important property of the requirements is that they are all *local* — no integration is involved in any of them, which makes constructing efficient strategies much easier. We now show that any efficient investment strategy satisfying these conditions induces an update rule of the desired competitive ratios. In particular, these requirements in combination ensure that the first dual constraint, the capacity constraint, is automatically satisfied.

Theorem 4.1. *Any efficient investment strategy λ_t induces primal variables x_t and y_t, such that all primal and dual constraints are satisfied, and the ratio between the primal and dual costs is bounded by* 2 *(resp. $e/(e-1)$) in the deterministic (resp. randomized) environment.*

To prove Theorem 4.1, we proceed by two steps. First we show the solutions are always feasible. In particular, we show the offline capacity constraint, absent from the requirements for efficient strategies, is always satisfied. This is captured by the following lemma.

Lemma 4.3. *Any efficient investment strategy λ_t satisfies for any $t \in \mathbb{R}_+$, $S \subseteq [n]$,*
$$\int_0^t \lambda_\tau(S)\, \mathrm{d}\tau = \int_0^t \sum_{i \in S} \lambda_\tau(i)\, \mathrm{d}\tau \le f(S).$$

We postpone the proof of the lemma, as well as all other missing proofs, to the appendix.

Then we bound the ratio between the primal and dual costs. The first part of the argument is exactly the same regardless of the environment being deterministic or randomized. The second part, on the other hand, is slightly different for the two environments. This is captured by the following lemma.

Lemma 4.4. *Let λ_t be an efficient investment strategy, and x_t and y_t be the primal variables induced in the deterministic (resp. randomized) environment. Then at any time $t \in \mathbb{R}_+$, the ratio between the primal cost and the dual cost,*
$$\left(\sum_S x_t(S) \cdot f(S) + \int_0^t \sum_S y_\tau(S) \cdot g_\tau(S)\, \mathrm{d}\tau\right) \Big/ \left(\int_0^t \lambda_\tau(i)\, \mathrm{d}\tau\right),$$
is no larger than 2 *(resp. $e/(e-1)$).*

It may appear counterintuitive that the same update rule for q_t may result in different ratios in the deterministic and randomized environments. To see why this is possible, recall that x_t and y_t depend only on p_t. In the deterministic environment, p_t is only allowed to be 0 or 1, and the relation between p_t and q_t is given by $p_t(i) = \mathbb{I}[q_t(i) = 1]$. On the other hand, in the randomized environment, p_t can be fractional, and the relation between p_t and q_t is $p_t(i) = (\exp(q_t(i)) - 1)/(e - 1)$. So even if q_t is the same, the primal variables x_t and y_t induced are different in the two environments, resulting in different ratios.

Theorem 4.1 is a direct corollary of Lemmas 4.3 and 4.4. With Theorem 4.1, it only remains to design an efficient investment strategy subject to local constraints.

4.4 The Order $\prec$

We now formally define the order $\prec$, and prove its properties, which enable Theorem 4.1 in a blackbox manner. Details of $\prec$ are necessary for the construction of our investment strategy.

We define $\prec$ in a more general case. That is, instead of restricting the third parameter to be additive (e.g., in $\prec_t = \prec(f, q_t, \lambda_t)$, λ_t can be viewed as an additive set function for any t), we allow it to be any submodular function. Fix f, q_t, and some submodular function g over the resources, let $\prec_t^g = \prec(f, q_t, g)$. In particular, $\prec_t = \prec_t^{\lambda_t}$. One may intuitively think of $\prec$ as induced by the leaderboard of a race. As discussed in the previous subsections, at time t, $q_t(i) < q_t(j)$ implies $i \prec_t^g j$. That is to say, i is ordered before j if i is strictly behind j in the race. It remains to define $\prec_t^g$ when $q_t(i) = q_t(j)$, i.e., i and j are currently at the same place. In such cases, it is natural to determine the order by the speeds of i and j, i.e., i is ordered before j if i is "slower" than j. We now formalize this intuition.

For any $\theta \in [0, 1]$, consider $S = S(=_t \theta)$ induced by q_t. Define $S(>_t \theta) = \{i \mid q_t(i) > \theta\}$. We say S is *ahead of* $S(<_t \theta)$, and *behind* $S(>_t \theta)$. In particular, the union of the three sets is $[n]$. For any $T \subseteq S$, define the *speed* of T to be

$$v_t(T) = \frac{g(T \mid S(<_t \theta) \cup (S \setminus T))}{f(T \mid S(>_t \theta))}.$$

Let the *leading subgroup* $L_t(S)$ be the largest[2] subset T of S with the highest speed, i.e., $v_t(T) = \max_{U \subseteq S} v_t(U)$. Then we define $\prec_t^g$ such that (1) for any $i, j \in L_t(S)$, $i \sim_t^g j$, and (2) for any $i \in S \setminus L_t(S)$ and $j \in L_t(S)$, $i \prec_t^g j$. That is, the leading subgroup is ordered after its complement in S, and within the leading subgroup, resources are tied to each other.

It remains to define $\prec_t^g$ within $S \setminus L_t(S)$. We do this recursively. In general, suppose U, V and W satisfy (1) $\forall i, j \in V$, $q_t(i) = q_t(j)$, (2) $\forall i \in U, j \in V$, $q_t(i) \le q_t(j)$, (3) $\forall i \in V, j \in W$, $q_t(i) \le q_t(j)$, (4) U, V, and W are pairwise disjoint, and (5) $U \cup V \cup W = [n]$. For $T \subseteq V$, let the speed of T in V between U and W be

$$v_t(T, U, V, W) = \frac{g(T \mid U \cup (V \setminus T))}{f(T \mid W)}.$$

Similarly, let the leading subgroup in V between U and W, $L_t(U, V, W)$, be the lexicographically smallest subset T of V such that $v_t(T, U, V, W) = \max_{X \subseteq S} v_t(X, U, V, W)$. Let $U_0 = S(<_t \theta)$, $V_0 = S$, $W_0 = S(>_t \theta)$. For $k \ge 0$, let $T_k = L_t(U_k, V_k, W_k)$, $U_{k+1} = U_k$, $V_{k+1} = V_k \setminus T_k$, $W_{k+1} = W_k \cup T_k$. Then clearly $\{T_0, T_1, \dots, \}$ form a partition of V_0. Moreover, T_k is non-empty as long as V_k is non-empty, and as a result $T_k = \emptyset$ for any $k > |V_0|$. For $i \in T_u$ and $j \in T_v$, (1) $i \prec_t^g j$ if $u > v$, (2) $j \prec_t^g i$ if $u < v$, and (3) $i \sim_t^g j$ if $u = v$. This completes the definition of $\prec$.

Now we state useful properties of $\prec$. First we show that when g is the investment rates λ_t, within each group T_k, all resources have the same speed $\frac{\mathrm{d}}{\mathrm{d}t} q_t(i)$, which is equal to the speed of the group.

Lemma 4.5 (same speed within a group). *Fix f, q_t and $g = \lambda_t$. Let $S = S(=_t \theta)$, $\{(U_k, V_k, W_k)\}_k$ be the sequence of partitions, and $\{T_k\}_k$ the sequence of leading subgroups induced by $(S(<_t \theta), S, S(>_t \theta))$. For any $k \ge 0$ and $i \in T_k$, $\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = v_t(T_k, U_k, V_k, W_k)$, as long as $T_k \neq \emptyset$.*

The following lemma shows that $\prec$ indeed breaks ties introduced by q_t according to the speeds.

Lemma 4.6 (tie-breaking from speeds). *Fix f, q_t and g. Let $S = S(=_t \theta)$, $\{(U_k, V_k, W_k)\}_k$ be the sequence of partitions, and $\{T_k\}_k$ the sequence of leading subgroups induced by $(S(<_t \theta), S, S(>_t \theta))$. For any $k \ge 0$, $v_t(T_k, U_k, V_k, W_k) > v_t(T_{k+1}, U_{k+1}, V_{k+1}, W_{k+1})$ as long as $T_{k+1} \neq \emptyset$.*

Now we are ready to prove the properties given in Section 4.2, i.e., Lemmas 4.1 and 4.2. Again we defer the proofs to the appendix.

[2] If there are multiple such subsets, let T be the lexicographically smallest one.

4.5 The Investment Strategy

Now we construct the investment rates λ_t given f and g_t, and show that the λ_t we construct satisfy all requirements of efficient strategies.

At time t, let $\prec_t^{g_t} = \prec(f, q_t, g_t)$. We construct λ_t for each group $S = S(=_t \theta)$ respectively. Fix $S = S(=_t \theta)$. If $\theta = 1$, then let $\lambda_t(i) = 0$ for all $i \in S$. Otherwise, consider sequence of partitions $\{(U_k, V_k, W_k)\}_k$ and leading groups $T_k = L_t(U_k, V_k, W_k)$ induced by (f, q_t, g_t) and $(S(<_t \theta), S, S(>_t \theta))$. For k such that $T_k \neq \emptyset$, let $\{\lambda_t(i)\}_{i\in T_k}$ be such that

$$\sum_{i\in T_k} \lambda_t(i) = g_t(T_k \mid U_k \cup (V_k \setminus T_k)),$$

$$\sum_{i\in S} \lambda_t(i) \leq \min\{g_t(S \mid U_k \cup (V_k \setminus T_k)), f(S \mid W_k) \cdot g(T_k \mid U_k \cup (V_k \setminus T_k))\}, \quad \forall S \subseteq T_k.$$

Performing the above for each $S(=_t \theta)$ and all leading groups induced yields investment rates $\{\lambda_t(i)\}_{i\in[n]}$ for all resources. The existence of such $\{\lambda_t(i)\}_{i\in T_k}$ within each T_k is guaranteed by the following lemma.

Lemma 4.7. *Fix $m \in \mathbb{N}$. Suppose $f_1, f_2 : 2^{[m]} \to \mathbb{R}_+$ are monotone submodular functions satisfying (1) $f_1(\emptyset) = f_2(\emptyset) = 0$, (2) $f_1([m]) = f_2([m]) = 1$, and (3) for any $S \subseteq [m]$, $f_1(S) + f_2([m] \setminus S) \geq 1$. Then there exist nonnegative $\{\lambda(i)\}_{i\in[m]}$ such that*

$$\sum_{i\in[n]} \lambda(i) = 1 \quad \text{and} \quad \sum_{i\in S} \lambda(i) \leq \min\{f_1(S), f_2(S)\}, \forall S \subseteq [m].$$

To see why the lemma implies the existence of $\{\lambda_t(i)\}_{i\in T_k}$, let $m = |T_k|$ and w.l.o.g. suppose T_k is the domain of f_1 and f_2. For any $S \subseteq T_k$, let

$$f_1(S) = \frac{f(S \mid W_k)}{f(T_k \mid W_k)} \quad \text{and} \quad f_2(S) = \frac{g_t(S \mid U_k \cup (V_k \setminus T_k))}{g_t(T_k \mid U_k \cup (V_k \setminus T_k))}.$$

It is easy to check $f_1(\emptyset) = f_2(\emptyset) = 0$ and $f_1(T_k) = f_2(T_k) = 1$. By the definition of T_k, for any $S \subseteq T_k$,

$$1 \geq \frac{v_t(S, U_k, V_k, W_k)}{v_t(T_k, U_k, V_k, W_k)} = \frac{1}{f_1(S)} - \frac{f_2(T_k \setminus S)}{f_1(S)}.$$

As a result, $f_1(S) + f_2(T_k \setminus S) \geq 1$, which is exactly the third condition required by Lemma 4.7. Let $\{\lambda(i)\}_{i\in T_k}$ be as guaranteed by Lemma 4.7. It is then easy to check that

$$\lambda_t(i) = \lambda(i) \cdot g_t(T_k \mid U_k \cup (V_k \setminus T_k))$$

gives $\{\lambda_t(i)\}_{i\in T_k}$ with the desired properties.

Now we prove that λ_t is in fact an efficient investment strategy, as desired. Recall that $\prec_t = \prec(f, q_t, \lambda_t)$. We first show that $\prec_t^{g_t}$ induced by g_t is exactly the same as $\prec_t$ induced by λ_t.

Lemma 4.8. *At any time $t \geq 0$, for any $i, j \in [n]$, $i \prec_t j \iff i \prec_t^{g_t} j$.*

The following lemma states that λ_t satisfies the piecewise constant condition.

Lemma 4.9. *For any $i \in [n]$, $\lambda_t(i)$ is piecewise constant in t.*

And finally, the following lemma states that λ_t satisfies the budget feasibility condition.

Lemma 4.10. *For any $t \in \mathbb{R}_+$ and $S \subseteq [n]$, $\sum_{i\in S} \lambda_t(i) \leq g_t(S)$.*

Finally, observe that λ_t clearly satisfies the no wasting condition. Together with Lemmas 4.9 and 4.10, this implies that λ_t constructed in this section is in fact an efficient investment strategy.

Theorem 4.2. *There exists an efficient investment strategy λ_t, satisfying the piecewise constant, budget feasibility, and no wasting conditions.*

Combined with Theorem 4.1, Theorem 4.2 directly implies the existence of competitive algorithms for (fractional) combinatorial ski rental and combinatorial online bipartite matching.

Theorem 4.3. *There exists*

- *a 2-competitive deterministic algorithm for combinatorial ski rental,*
- *an online $e/(e-1)$-competitive solution to the LP formulation of combinatorial ski rental, and*
- *an $e/(e-1)$-competitive algorithm for combinatorial online bipartite matching.*

4.6 Rounding Scheme with Multiple Resources

In this section, we give a rounding scheme, which, given a fractional primal solution, yields an integral primal solution preserving the cost of the fractional solution in expectation. Together with Theorems 4.1 and 4.2, this implies an $e/(e-1)$-competitive randomized algorithm for the combinatorial ski rental problem.

The rounding scheme is simple and similar to the one presented in Section 3. Again, we draw a number r uniformly at random from $[0, 1]$, at the very beginning of the algorithm. We upgrade the set purchased to include resource i as soon as $q_t(i)$ becomes at least r. We now show that this rounding scheme preserves the cost in expectation.

Theorem 4.4. *Fix f, g_t, q_t, and x_t induced by q_t, and let S_t be the set purchased at time t, given by the above rounding scheme. At any time $t \in \mathbb{R}_+$, we have*

$$\mathbb{E}\left[f(S_t) + \int_0^t g_\tau([n] \setminus S_\tau)\, \mathrm{d}\tau\right] = \sum_S x_t(S) \cdot f(S) + \int_0^t \sum_S x_\tau(S) \cdot g_\tau([n] \setminus S)\, \mathrm{d}\tau.$$

As a result, there is an $e/(e-1)$-competitive randomized algorithm for combinatorial ski rental.

5 HARDNESS RESULTS

We prove in this section that for combinatorial ski rental, the restrictions we put on the cost functions, i.e., (1) the cost of purchasing f being submodular, (2) the cost of renting g_t being submodular, and (3) the cost of purchasing is for upgrading, rather than separate purchase, are all necessary to allow for a constant competitive ratio.

Theorem 5.1. *No (possibly randomized and / or inefficient) algorithm is $o(\sqrt{\log n})$-competitive for combinatorial ski rental when the purchasing cost f is allowed to be XOS, even if the renting cost is additive.*

Theorem 5.2. *No (possibly randomized and / or inefficient) algorithm is $o(\log n)$-competitive for combinatorial ski rental when the renting cost g_t is allowed to be supermodular, even if the purchasing cost is additive.*

Theorem 5.3. *No (possibly randomized and / or inefficient) algorithm is $O(n^{1-\varepsilon})$-competitive for combinatorial ski rental when upgrading is not allowed*[3]*, for any $\varepsilon > 0$, even if the purchasing cost is submodular and the renting cost is additive.*

ACKNOWLEDGMENTS

We are thankful for support from NSF under award IIS-1814056. We also thank Debmalya Panigrahi for helpful discussion, and anonymous reviewers for helpful feedback.

[3] That is, the total cost incurred by the algorithm is $\sum_t f(\Delta S_t) + \int_t g_t([n] \setminus S_t)$.

REFERENCES

[1] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1253–1264.

[2] Noga Alon, Baruch Awerbuch, and Yossi Azar. 2003. The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, 100–105.

[3] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Seffi Naor. 2006. A general approach to online network optimization problems. *ACM Transactions on Algorithms (TALG)* 2, 4 (2006), 640–660.

[4] Baruch Awerbuch and Yossi Azar. 1997. Buy-at-bulk network design. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, 542–547.

[5] Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalya Panigrahi. 2013. Online mixed packing and covering. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 85–100.

[6] Yossi Azar, Niv Buchbinder, TH Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, et al. 2016. Online algorithms for covering and packing problems with convex objectives. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 148–157.

[7] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. 2010. Towards the randomized k-server conjecture: A primal-dual approach. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 40–55.

[8] Nikhil Bansal, Niv Buchbinder, and Joseph Seffi Naor. 2008. Randomized competitive algorithms for generalized caching. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, 235–244.

[9] Nikhil Bansal, Niv Buchbinder, and Joseph Seffi Naor. 2012. A primal-dual randomized algorithm for weighted paging. *Journal of the ACM (JACM)* 59, 4 (2012), 19.

[10] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*. Springer, 253–264.

[11] Niv Buchbinder and Joseph Naor. 2006. Improved bounds for online routing and packing via a primal-dual approach. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 293–304.

[12] Niv Buchbinder and Joseph Seffi Naor. 2009. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends® in Theoretical Computer Science* 3, 2–3 (2009), 93–263.

[13] TH Hubert Chan, Zhiyi Huang, Shaofeng H-C Jiang, Ning Kang, and Zhihao Gavin Tang. 2017. Online Submodular Maximization with Free Disposal: Randomization Beats for Partition Matroids. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1204–1223.

[14] Nikhil R Devanur and Thomas P Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*. ACM, 71–78.

[15] Nikhil R Devanur, Zhiyi Huang, Nitish Korula, Vahab S Mirrokni, and Qiqi Yan. 2016. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Transactions on Economics and Computation (TEAC)* 4, 3 (2016), 14.

[16] Nikhil R Devanur, Kamal Jain, and Robert D Kleinberg. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 101–107.

[17] Daniel R Dooly, Sally A Goldman, and Stephen D Scott. 1998. TCP dynamic acknowledgment delay: theory and practice. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 389–398.

[18] Uriel Feige. 2009. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.* 39, 1 (2009), 122–142.

[19] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. 2010. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*. Springer, 182–194.

[20] Jon Feldman, Nitish Korula, Vahab Mirrokni, S Muthukrishnan, and Martin Pál. 2009. Online ad assignment with free disposal. In *International workshop on internet and network economics*. Springer, 374–385.

[21] Gagan Goel and Aranyak Mehta. 2008. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 982–991.

[22] Sreenivas Gollapudi and Debmalya Panigrahi. 2019. Online Algorithms for Rent-Or-Buy with Expert Advice. In *International Conference on Machine Learning*. 2319–2327.

[23] Elad Hazan and Satyen Kale. 2012. Online submodular minimization. *Journal of Machine Learning Research* 13, Oct (2012), 2903–2922.

[24] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. 2018. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 17–29.

[25] Zhiyi Huang and Anthony Kim. 2018. Welfare maximization with production costs: A primal dual approach. *Games and Economic Behavior* (2018).

[26] Zhiyi Huang, Binghui Peng, Zhihao Gavin Tang, Runzhou Tao, Xiaowei Wu, and Yuhao Zhang. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2875–2886.

[27] Michael Kapralov, Ian Post, and Jan Vondrák. 2013. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1216–1225.

[28] Anna R Karlin, Claire Kenyon, and Dana Randall. 2001. Dynamic TCP acknowledgement and other stories about $e/(e-1)$. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM, 502–509.

[29] Anna R Karlin, Mark S Manasse, Lyle A McGeoch, and Susan Owicki. 1994. Competitive randomized algorithms for nonuniform problems. *Algorithmica* 11, 6 (1994), 542–571.

[30] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. ACM, 352–358.

[31] Zvi Lotker, Boaz Patt-Shamir, and Dror Rawitz. 2012. Rent, lease, or buy: Randomized algorithms for multislope ski rental. *SIAM Journal on Discrete Mathematics* 26, 2 (2012), 718–736.

[32] Aleksander Madry and Debmalya Panigrahi. 2011. The Semi-stochastic Ski-rental Problem. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[33] Aranyak Mehta. 2013. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science* 8, 4 (2013), 265–368.

[34] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2005. Adwords and generalized on-line matching. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE, 264–273.

[35] Yajun Wang and Sam Chiu-wai Wong. 2015. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *International Colloquium on Automata, Languages, and Programming*. Springer, 1070–1081.

[36] Yajun Wang and Sam Chiu-wai Wong. 2016. Matroid online bipartite matching and vertex cover. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 437–454.

A OMITTED PROOFS IN SECTION 4.3

Proof of Lemma 4.3. Consider any set $S \subseteq [n]$ of resources. We need to show that at any moment t,

$$\int_0^t \sum_{i \in S} \lambda_\tau(i)\,\mathrm{d}\tau \leq f(S).$$

We prove the above by showing, if equality holds between the two sides of the above inequality, then it must be the case that for any $i \in S$, $q_t(i) = 1$, and therefore $\lambda_t(i) = 0$ by the no wasting requirement. As we will show, this is a corollary of the following fact.

Lemma A.1 (monotonicity in investment rates). *Let λ_t and λ'_t be two piecewise constant investment strategies, satisfying for any $i \in [n]$, $t \in \mathbb{R}_+$, $\lambda_t(i) \geq \lambda'_t(i)$. Let q_t and q'_t be the corresponding investments induced by the two strategies respectively. Then for any $i \in [n]$, $t \in \mathbb{R}_+$, we have $q_t(i) \geq q'_t(i)$.*

The lemma can be viewed as monotonicity of the mapping from an investment strategy to the resulting group of investments. It essentially says, that if one investment strategy is pointwise dominated by another strategy, then the corresponding group of investments is also pointwise dominated by that induced by the other strategy. We now prove the lemma.

Proof of Lemma A.1. Observe that q_t and q'_t are continuous and piecewise linear. Therefore we only need to show that, any time t, if i satisfies $q_t(i) - q'_t(i) = \min_j(q_t(j) - q'_t(j))$, then

$$\frac{\lambda_t(i)}{h_t(i)} \geq \frac{\lambda'_t(i)}{h'_t(i)}.$$

This is a direct corollary of the monotone rates property (Lemma 4.2), since for any j, $q_t(j) - q_t(i) \geq q'_t(j) - q'_t(i)$, so if $q'_t(j) - q'_t(i)$ is positive (resp. nonnegative), then so is $q_t(j) - q_t(i)$. To see why this implies the lemma, suppose at time t_1,

$$\min_j(q_t(j) - q'_t(j)) < 0.$$

Let

$$t_0 = \sup\{t \le t_1 \mid \min_j(q_t(j) - q'_t(j)) \ge 0\}.$$

By continuity we have

$$\min_j(q_{t_0}(j) - q'_{t_0}(j)) = 0.$$

Since q_t and q'_t are piecewise linear, there exists i and $\varepsilon > 0$, such that for any $t \in (t_0, t_0 + \varepsilon]$,

$$q_t(i) - q'_t(i) = \min_j(q_t(j) - q'_t(j)) < 0.$$

But then on $(t_0, t_0 + \varepsilon]$,

$$\frac{\mathrm{d}}{\mathrm{d}t}(q_t(i) - q'_t(i)) = \frac{\lambda_t(i)}{h_t(i)} - \frac{\lambda'_t(i)}{h'_t(i)} \ge 0,$$

and in particular,

$$0 > q_{t_0+\varepsilon}(i) - q'_{t_0+\varepsilon}(i) = q_{t_0}(i) - q'_{t_0}(i) + \int_{t_0}^{t_0+\varepsilon} \frac{\mathrm{d}}{\mathrm{d}t}(q_t(i) - q'_t(i))\,\mathrm{d}t \ge 0,$$

a contradiction. □

Given Lemma A.1, we relax the actual strategy λ_t to one dominated by λ_t. Formally, we define a dominated strategy λ'_t, such that

$$\lambda'_t(i) = \begin{cases} \lambda_t(i), & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases}.$$

Note that λ'_t is piecewise constant if λ_t is piecewise constant. Let q'_t be the investments induced by λ'_t. Now all we need to show is, whenever

$$\int_0^t \sum_{i \in S} \lambda'_\tau(i)\,\mathrm{d}\tau = f(S),$$

we always have $q'_t(i) = 1$ for all $i \in S$. To see why this is true, we define a investment distribution z'_t over subsets of resources, which is determined by the investments q'_t in the same way that x_t is determined by p_t. That is, for any $T \subseteq [n]$,

$$z'_t(T) = \Pr_{\theta \sim \mathcal{U}(0,1)}[T = \{i \mid q'_t(i) \ge \theta\}].$$

Consider the cost of purchasing the investment distribution z'_t and the amount by which it increases between time t and $t + \mathrm{d}t$. Let $\prec'_t = \prec(f, q'_t, \lambda'_t)$. Let $S(\succ'_t i) = \{j \mid j \succ'_t i\}$, and $S(\sim'_t i)$ and $S(\prec'_t i)$ similarly defined. Recall that the unit price for i induced by (f, q'_t, λ'_t) is

$$h'_t(i) = f(S(\sim'_t i) \mid S(\succ'_t i)) \cdot \frac{\lambda'_t(i)}{\lambda'_t(S(\sim'_t i))}.$$

Observe that at time t, the expected cost of purchasing a set of resources distributed according to z'_t can be written as

$$\sum_{T \subseteq [n]} z'_t(T) \cdot f(T) = \sum_{i \in [n]} q'_t(i) \cdot h'_t(i).$$

Suppose $<'_t$ does not change at time t. The increase of the cost, calculated in a somewhat non-rigorous but more informative way, is then

$$\begin{aligned}
&\left(\sum_{T\subseteq[n]} z'_{t+\mathrm{d}t}(T)\cdot f(T)\right)-\left(\sum_{T\subseteq[n]} z'_t(T)\cdot f(T)\right)\\
&=\sum_{i\in[n]}\left(q'_{t+\mathrm{d}t}(i)\cdot h'_{t+\mathrm{d}t}(i)-q'_t(i)\cdot h'_t(i)\right) && \text{(expanding the cost as a telescoping sum)}\\
&=\sum_{i\in[n]}(q'_{t+\mathrm{d}t}(i)-q'_t(i))\cdot h'_t(i)\\
& && \text{($<'_t$ changes only on a zero-measure set, so $h'_t(i)$ is constant in t almost everywhere)}\\
&=\sum_{i\in[n]}\frac{\lambda'_t(i)\,\mathrm{d}t}{h'_t(i)}\cdot h'_t(i) && \text{(dynamics of the market)}\\
&=\sum_{i\in[n]}\lambda'_t(i)\,\mathrm{d}t\\
&=\sum_{i\in S}\lambda_t(i)\,\mathrm{d}t. && \text{(definition of λ')}
\end{aligned}$$

So the increase of the cost of z'_t is exactly the amount of budget invested in S. In other words, almost everywhere,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{T\subseteq[n]} z'_t(T)\cdot f(T)\right)=\sum_{i\in S}\lambda_t(i).$$

As a result, we have

$$\sum_{T\subseteq[n]} z'_t(T)\cdot f(T)=\int_0^t\sum_{i\in S}\lambda_\tau(i)\,\mathrm{d}\tau=f(S).$$

Now since the cost of purchasing z'_t is clearly monotone in z'_t and $q'_t(i)=0$ for any $i\notin S$, the only distribution that has cost $f(S)$ is $z'_t(S)=1$ and $z'_t(T)=0$ for any $T\neq S$. This investment distribution corresponds uniquely to the group of investments where $q'_t(i)=1$ for all $i\in S$ and $q'_t(i)=0$ otherwise. By Lemma A.1, this implies for all $i\in S$, $q_t(i)=1$, which is exactly our desired condition. This finishes the proof for feasibility. □

Proof of Lemma 4.4. We now bound the ratio between the primal and dual costs, in the deterministic and randomized environments respectively.

The ratio in the deterministic case. Similar to the single-resource analysis, we decompose the primal cost into two parts, the cumulative cost of purchasing and that of renting. We argue that each part is no larger than the dual cost at any time t. Consider first the cost of purchasing,

$$\sum_S x_t(S)\cdot f(S)=\sum_i p_t(i)\cdot h_t(i).$$

Again, let z_t be the investment distribution induced by q_t in the same way that x_t is induced by p_t. As in the proof of Lemma A.1, the dual cost can be written as

$$\int_0^t\sum_i\lambda_\tau(i)\,\mathrm{d}\tau=\sum_S z_t(S)\cdot f(S)=\sum_i q_t(i)\cdot h_t(i).$$

Now recall that in the deterministic environment, $p_t(i) = \mathbb{I}[q_t(i) = 1] \leq q_t(i)$. It follows directly that

$$\sum_i p_t(i) \cdot h_t(i) \leq \sum_i q_t(i) \cdot h_t(i).$$

In other words, the primal cost of purchasing is upper bounded by the dual cost.

Now consider the primal cost of renting. We show that at any time t, the renting cost and the dual cost increase at the same rate. We incur cost of rate $g_t(S(<_t 1))$ where $S(<_t 1)$ is the set of all resources not yet fully purchased, i.e., $S(<_t 1) = \{i \mid q_t(i) < 1\}$. The rate can be written as

$$g_t(S(<_t 1)) = \sum_{0 \leq \theta < 1 : \exists i, q_t(i) = \theta} g_t(S(=_t \theta) \mid S(<_t \theta)).$$

On the other hand, the increase of the dual cost satisfies the group-wise full spending property. This implies that the rate at which the dual cost increases can be written as

$$\sum_i \lambda_t(i) = \sum_{0 \leq \theta < 1 : \exists i, q_t(i) = \theta} \sum_{j \in S(=_t \theta)} \lambda_t(j) = \sum_{0 \leq \theta < 1 : \exists i, q_t(i) = \theta} g_t(S(=_t \theta) \mid S(<_t \theta)).$$

This is precisely the primal rate of increase of the renting cost. We therefore conclude that the primal cost is always no larger than twice the dual cost.

The ratio in the randomized case. Recall that in the randomized case, $p_t(i) = \frac{1}{e-1}(\exp(q_t(i)) - 1)$. We consider directly the rate at which the primal cost increases, which, given Lemma 4.1, can be written as

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_S x_t(S) \cdot f(S)\right) + \sum_S y_t(S) \cdot g_t(S) \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_S x_t(S) \cdot f(S)\right) + \sum_S x_t([n] \setminus S) \cdot g_t(S) && (y_t(S) = x_t([n] \setminus S)) \\
&= \sum_i \left(\frac{\mathrm{d}}{\mathrm{d}t} p_t(i)\right) \cdot h_t(i) + \sum_S x_t([n] \setminus S) \cdot g_t(S) && (h_t(i) \text{ is constant almost everywhere}) \\
&= \sum_i \left(\frac{1}{e-1} \exp(q_t(i)) \cdot \frac{\mathrm{d}}{\mathrm{d}t} q_t(i)\right) \cdot h_t(i) + \sum_i g_t(\{i\} \mid S(<_t i) \cup (S(\sim_t i) \cap [i-1]))(1 - p_t(i)) \\
& && \text{(expanding second term as a telescoping sum)} \\
&= \sum_i \left(\frac{1}{e-1} \exp(q_t(i)) \cdot \frac{\lambda_t(i)}{h_t(i)}\right) \cdot h_t(i) + \sum_{0 \leq \theta \leq 1 : \exists i, 1 - p_t(i) = \theta} g_t(S(=_t \theta) \mid S(<_t \theta)) \cdot \theta \\
& && \text{(grouping by } p_t(i)) \\
&= \sum_i \left(p_t(i) + \frac{1}{e-1}\right) \cdot \lambda_t(i) + \sum_{0 \leq \theta \leq 1 : \exists i, 1 - p_t(i) = \theta} \sum_{j \in S(=_t \theta)} \lambda_t(j) \cdot (1 - p_t(j)) \\
& && \text{(group-wise full spending)} \\
&= \sum_i \left(p_t(i) + \frac{1}{e-1}\right) \cdot \lambda_t(i) + \sum_i \lambda_t(i) \cdot (1 - p_t(i)) \\
&= \frac{e}{e-1} \sum_i \lambda_t(i).
\end{aligned}$$

On the other hand, the dual cost increases at rate exactly $\sum_i \lambda_t(i)$. This establishes the desired ratio of $e/(e-1)$. □

B OMITTED PROOFS IN SECTION 4.4

Proof of Lemma 4.5. Recall that by the market dynamics,

$$\frac{\mathrm{d}}{\mathrm{d}t}q_t(i) = \frac{\lambda_t(i)}{h_t(i)} = \lambda_t(i)\cdot\left(f(T_k \mid W_k)\cdot\frac{\lambda_t(i)}{\lambda_t(W_k)}\right)^{-1}$$
$$= \frac{\lambda_t(W_k)}{f(T_k \mid W_k)} = \frac{\lambda_t(W_k \mid U_k \cup (V_k \setminus T_k))}{f(T_k \mid W_k)} = v_t(T_k, U_k, V_k, W_k). \qquad \square$$

Proof of Lemma 4.6. Suppose otherwise, i.e.,

$$v_t(T_k, U_k, V_k, W_k) \le v_t(T_{k+1}, U_{k+1}, V_{k+1}, W_{k+1}).$$

We show that

$$v_t(T_k \cup T_{k+1}, U_k, V_k, W_k) \ge v_t(T_k, U_k, V_k, W_k),$$

contradicting the definition of T_k, since $T_k \cup T_{k+1}$ is larger than T_k and has speed no smaller than T_k. In fact,

$$\begin{aligned} v_t(T_k \cup T_{k+1}, U_k, V_k, W_k) &= \frac{g(T_k \cup T_{k+1} \mid U_k \cup (V_k \setminus (T_k \cup T_{k+1})))}{f(T_k \cup T_{k+1} \mid W_k)} \\ &= \frac{g(T_k \mid U_k \cup (V_k \setminus T_k)) + g(T_{k+1} \mid U_k \cup (V_k \setminus (T_k \cup T_{k+1})))}{f(T_k \mid W_k) + f(T_{k+1} \mid W_k \cup T_k)} \\ &= \frac{g(T_k \mid U_k \cup (V_k \setminus T_k)) + g(T_{k+1} \mid U_{k+1} \cup (V_{k+1} \setminus T_{k+1}))}{f(T_k \mid W_k) + f(T_{k+1} \mid W_{k+1})}. \\ & \qquad (U_{k+1} = U_k,\ V_{k+1} = V_k \setminus T_k,\ W_{k+1} = W_k \cup T_k) \end{aligned}$$

Observe that

$$\begin{aligned} v_t(T_k, U_k, V_k, W_k) &= \frac{g(T_k \mid U_k \cup (V_k \setminus T_k))}{f(T_k \mid W_k)}, \\ v_t(T_{k+1}, U_{k+1}, V_{k+1}, W_{k+1}) &= \frac{g(T_{k+1} \mid U_{k+1} \cup (V_{k+1} \setminus T_{k+1}))}{f(T_{k+1} \mid W_{k+1})}. \end{aligned}$$

As a result,

$$v_t(T_k \cup T_{k+1}, U_k, V_k, W_k) \ge v_t(T_k, U_k, V_k, W_k) = \max_X v_t(X, U_k, V_k, W_k),$$

a contradition. This finishes the proof. □

Proof of Lemma 4.1. Fix f, and λ_t, which together determine q_t. Let $\prec_t = \prec(f, q_t, \lambda_t)$. Given continuity of λ_t almost everywhere, we only need to show that for resources i and j where $q_t(i) = q_t(j)$,

$$i \succ_t j \Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t}q_t(i) \ge \frac{\mathrm{d}}{\mathrm{d}t}q_t(j),$$

and

$$i \sim_t j \Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t}q_t(i) = q_t(j),$$

if $\lambda_t(i)$ and $\lambda_t(j)$ are both constant at t. The lemma then follows from a standard ε-δ argument, which implies at any time t when $\lambda_t(i)$ is constant for all i, there exists $\varepsilon > 0$ such that $\prec_t = \prec_{t'}$ for any $t' \in [t, t+\varepsilon)$.

Again, consider $S = S(=_t \theta)$ which contains i and j, and let $\{(U_k, V_k, W_k)\}_k$ and $\{T_k\}_k$ be the sequences of partitions and leading subgroups. Suppose $i \in T_u$ and $j \in T_v$. When $i \succ_t j$, by the definition of $\prec_t$, $u < v$. By Lemmas 4.5 and 4.6, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = v_t(T_u, U_u, V_u, W_u) \geq v_t(T_v, U_v, V_v, W_v) = \frac{\mathrm{d}}{\mathrm{d}t} q_t(j).$$

When $i \sim_t j$, by the definition of $\prec_t$, $u = v$. By Lemma 4.5, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = v_t(T_u, U_u, V_u, W_u) = v_t(T_v, U_v, V_v, W_v) = \frac{\mathrm{d}}{\mathrm{d}t} q_t(j). \qquad \square$$

Proof of Lemma 4.2. Fix i, f, q_t, q'_t, λ_t, λ'_t as in the lemma. Let $(U_0, V_0, W_0) = (S(<_t q_t(i)), S(=_t q_t(i)), S(>_t q_t(i))$ induced by q_t and $(U'_0, V'_0, W'_0) = (S(<'_t q'_t(i)), S(='_t q'_t(i)), S(>'_t q'_t(i))$ induced by q'_t, where $W_0 \supseteq W'_0$ and $V_0 \cup W_0 \supseteq V'_0 \cup W'_0$. Moreover, let $\{(U_k, V_k, W_k)\}_k$ and $\{(U'_k, V'_k, W'_k)\}_k$ be the sequences of partitions generated by (U_0, V_0, W_0) and λ_t, and (U'_0, V'_0, W'_0) and λ'_t, respectively. Let $\{T_k\}_k$ and $\{T'_k\}_k$ be the respective sequences of leading groups.

Suppose $i \in T_u$ and $i \in T'_v$. Let w be the smallest integer such that $T'_w \cap V_u \neq \emptyset$. Observe that $w \leq v$, since $i \in T'_v \cap T_u \subseteq T'_v \cap V_u$. We now show that

$$v_t(T'_w \cap V_u, U_u, V_u, W_u) \geq v'_t(T'_w, U'_w, V'_w, W'_w).$$

Observe first that $W_u \supseteq W'_w$. This is because W'_w consists of two parts: $W'_0 \subseteq W_0$, and $\bigcup_{\ell < w} T'_\ell \subseteq V'_0 \subseteq V_0 \cup W_0$. Also, the latter part does not intersect V_u by the choice of w, which means it must be contained in $W_u = V_0 \cup W_0 \setminus V_u$. As a result, $W_u \supseteq W'_w$. Now observe that $T'_w \setminus V_u = T'_w \cap W_u \subseteq W_u$. This is simply because $T'_w \subseteq V'_0 \subseteq V_0 \cup W_0$. Now by optimality of T'_w, it must be the case that

$$\begin{aligned}\frac{\lambda'_t(T'_w \cap V_u)}{f(T'_w \cap V_u \mid W'_w \cup (T'_w \setminus V_u))} &\geq \frac{\lambda'_t(T'_w \cap V_u) + \lambda'_t(T'_w \setminus V_u)}{f(T'_w \cap V_u \mid W'_w \cup (T'_w \setminus V_u)) + f(T'_w \setminus V_u \mid W'_w)} \\ &= v'_t(T'_w, U'_w, V'_w, W'_w),\end{aligned}$$

since otherwise $v'_t(T'_w \setminus V_u, U'_w, V'_w, W'_w)$ would be strictly larger than $v'_t(T'_w, U'_w, V'_w, W'_w)$. On the other hand, by submodularity of f and since $W_u \supseteq W'_w$ and $T'_w \setminus V_u \subseteq W_u$,

$$v_t(T'_w \cap V_u, U_u, V_u, W_u) = \frac{\lambda_t(T'_w \cap V_u)}{f(T'_w \cap V_u \mid W_u)} \geq \frac{\lambda'_t(T'_w \cap V_u)}{f(T'_w \cap V_u \mid W'_w \cup (T'_w \setminus V_u))} \geq v'_t(T'_w, U'_w, V'_w, W'_w).$$

Given the above inequality, we have

$$\begin{aligned}\frac{\lambda_t(i)}{h_t(i)} &= v_t(T_u, U_u, V_u, W_u) && \text{(Lemma 4.5)} \\ &\geq v_t(T'_w \cap V_u, U_u, V_u, W_u) && \text{(definiton of } T_u\text{)} \\ &\geq v'_t(T'_w, U'_w, V'_w, W'_w) \\ &\geq v'_t(T'_v, U'_v, V'_v, W'_v) && \text{(Lemma 4.6 and } w \leq v\text{)} \\ &= \frac{\lambda'_t(i)}{h'_t(i)}. && \text{(Lemma 4.5)}\end{aligned}$$

This concludes the proof of the lemma. □

C OMITTED PROOFS IN SECTION 4.5

Proof of Lemma 4.7. Consider the following LP,

$$\begin{aligned} \max \quad & \sum_{i \in [m]} \lambda(i) \\ \text{s.t.} \quad & \sum_{i \in S} \lambda(i) \le f_1(S) \quad \forall S \subseteq [m] \\ & \sum_{i \in S} \lambda(i) \le f_2(S) \quad \forall S \subseteq [m] \\ & \lambda(i) \ge 0 \qquad\qquad \forall i \in [m] \end{aligned}$$

and its dual,

$$\begin{aligned} \min \quad & \sum_{S \subseteq [m]} (\alpha_S f_1(S) + \beta_S f_2(S)) \\ \text{s.t.} \quad & \sum_{S \ni i} (\alpha_S + \beta_S) \ge 1 \qquad \forall i \in [m] \\ & \alpha_S, \beta_S \ge 0 \qquad\qquad \forall S \subseteq [m] \end{aligned}$$

The plan is to show that the optimal value for the primal LP is at least (and in fact, precisely) 1, and the $\lambda(i)$ which induce this value then satisfy the conditions of the lemma. In order to establish this, we lower bound by 1 the value of the dual objective induced by any feasible solution.

Let $(\{\alpha_S\}, \{\beta_S\})$ be any feasible dual solution. We first show that we can modify the solution into a hierarchical form, while not increasing the objective. Let $u(i) = \sum_{S \ni i} \alpha_S$, $u(0) = 1$ and $u(m+1) = 0$. W.l.o.g., suppose $u(i) \le u(i-1)$ for $i \in [m]$. Define $\{\alpha'_S\}_S$ such that $\alpha'_{[i]} = u(i) - u(i+1)$ for $i \in [m]$ and $\alpha'_S = 0$ otherwise. We argue that replacing $\{\alpha_S\}_S$ with $\{\alpha'_S\}_S$ keeps the solution feasible and does not increase the objective. In fact, let $r_i = f(\{i\} \mid [i-1])$. We have

$$\begin{aligned} \sum_S \alpha'_S f_1(S) &= \sum_{i \in [m]} \alpha'_{[i]} f_1([i]) = \sum_{i \in [m]} (u(i) - u(i+1)) \sum_{j \le i} r_j \\ &= \sum_{j \in [m]} r_j \sum_{i \ge j} (u(i) - u(i+1)) = \sum_{j \in [m]} r_j \cdot u(j). \end{aligned}$$

On the other hand, for any S, by submodularity of f_1,

$$f_1(S) = \sum_{i \in S} f(\{i\} \mid S \cap [i-1]) \le \sum_{i \in S} f(\{i\} \mid [i-1]) = \sum_{i \in S} r_i.$$

So,

$$\sum_S \alpha_S f_1(S) \ge \sum_S \alpha_S \sum_{i \in S} r_i = \sum_{i \in [m]} r_i \sum_{S \ni i} \alpha_S = \sum_{i \in [m]} r_i \cdot u(i) = \sum_S \alpha'_S f_1(S),$$

as desired. So we may assume w.l.o.g. that $\{\alpha_S\}_S$ are induced by $u(i)$. Similarly, let $v(i) = \sum_{S \ni i} \beta_S$. We may assume that $\{\beta_S\}_S$ are induced by $v(i)$. Moreover, we may assume $u(i) + v(i) = 1$ for any $i \in [m]$ for optimality.

Now consider the objective. Again, w.l.o.g., suppose $u(0) = 1$, $u(m+1) = 0$, and $u(i) \le u(i-1)$ for $i \in [m]$. As a result, $v(i) \ge v(i-1)$ for $i \in [m]$. And for any $i \in [m]$,

$$\alpha_{[i]} = \beta_{[m] \setminus [i]} = u(i) - u(i+1).$$

The objective then can be written as

$$\begin{aligned}\sum_S(\alpha_S f_1(S) + \beta_S f_2(S)) &= \sum_i(\alpha_{[i]} f_1([i]) + \beta_{[m]\setminus[i]} f_2([m] \setminus [i])) \\ &= \sum_{0\le i\le m} (u(i) - u(i+1))(f_1([i]) + f_2([m] \setminus [i])) \\ &\ge \sum_{0\le i\le m} (u(i) - u(i+1)) \qquad (f_1(S) + f_2([m] \setminus S) \ge 1 \text{ for all } S) \\ &= u(0) - u(m+1) \\ &= 1.\end{aligned}$$

In other words, for any feasible dual solution, the objective value is at least 1. This concludes the proof of the lemma. □

Proof of Lemma 4.8. Consider group $S = S(=_t \theta)$. Let $\{(U_k, V_k, W_k)\}_k$ be the sequence of partitions, and $\{T_k\}_k$ the sequence of leading subgroups induced by (f, q_t, g_t) and $(S(<_t \theta), S, S(>_t \theta))$. We show that (f, q_t, λ_t) and $(S(<_t \theta), S, S(>_t \theta))$ induce exactly the same sequence of partitions. Let v_t and L_t be the speeds and leading groups induced by g_t, and v'_t and L'_t those induced by λ_t. Fix some k. For any $T \subseteq V_k$, we show that (1) $v'_t(T, U_k, V_k, W_k) \le v_t(T_k, U_k, V_k, W_k)$, and (2) $T_k \subsetneq T \implies v'_t(T, U_k, V_k, W_k) < v_t(T_k, U_k, V_k, W_k)$. Let $X_\ell = T \cap T_\ell$ for $\ell \ge k$. Clearly $T = \bigcup_{\ell\ge k} X_\ell$. For each ℓ where $X_\ell \ne \emptyset$, by the construction of λ_t,

$$v_t(T_\ell, U_\ell, V_\ell, W_\ell) \ge v_t(X_\ell, U_\ell, V_\ell, W_\ell) = \frac{\lambda_t(X_\ell)}{f(X_\ell \mid W_\ell)}.$$

And by submodularity of f,

$$v_t(T_\ell, U_\ell, V_\ell, W_\ell) \ge \frac{\lambda_t(X_\ell)}{f(X_\ell \mid W_\ell)} \ge \frac{\lambda_t(X_\ell)}{f(X_\ell \mid (T \cap W_\ell) \cup W_k)}.$$

Now consider $v'_t(T \mid U_k \cup (V_k \setminus T))$. We have

$$v'_t(T \mid U_k \cup (V_k \setminus T)) = \frac{\lambda_t(T)}{f(T \mid W_k)} = \frac{\sum_{\ell\ge k} \lambda_t(X_\ell)}{\sum_{\ell\ge k} f(X_\ell \mid (T \cap W_\ell) \cup W_k)}.$$

On the other hand, we know that,

$$\frac{\lambda_t(X_k)}{f(X_k \mid W_k)} \le v_t(T_k, U_k, V_k, W_k),$$

and for any $\ell > k$ where $X_\ell \ne \emptyset$, by Lemma 4.6,

$$\frac{\lambda_t(X_\ell)}{f(X_\ell \mid (T \cap W_\ell) \cup W_k)} \le v_t(T_\ell, U_\ell, V_\ell, W_\ell) < v_t(T_k, U_k, V_k, W_k).$$

So we have

$$v'_t(T, U_k, V_k, W_k) \le v_t(T_k, U_k, V_k, W_k),$$

where equality holds only when $T \subseteq T_k$.

And for $T_k = L_t(U_k, V_k, W_k)$, we have

$$\lambda_t(T_k) = g_t(T_k \mid U_k \cup (V_k \setminus T)).$$

So

$$v_t(T_k, U_k, V_k, W_k) = v'_t(T_k, U_k, V_k, W_k).$$

Now we know the new highest speed is previsely the old highest speed, which is again achieved by T_k, or possibly its subsets. By the definition of the leading group, T_k is still the leading group induced by λ_t, i.e.,

$$L_t(U_k, V_k, W_k) = T_k = L'_t(U_k, V_k, W_k).$$

A simple induction then implies (f, q_t, λ_t) generate exactly the same sequence of partitions, and $\prec_t$ and $\prec_t^{g_t}$ are in fact the same order. □

Proof of Lemma 4.9. We only need to show that $\prec_t^{g_t}$ is constant almost everywhere. By Lemmas 4.5, 4.6 and 4.8, whenever g_t is constant, for i and j where $q_t(i) = q_t(j)$,

$$i \prec_t^{g_t} j \Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t} q_t(i) \le \frac{\mathrm{d}}{\mathrm{d}t} q_t(j),$$

and

$$i \sim_t^{g_t} j \Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t} q_t(i) = \frac{\mathrm{d}}{\mathrm{d}t} q_t(j).$$

Now by the same argument as in the proof of Lemma 4.1, $\prec_t^{g_t}$ is constant almost everywhere, and the lemma follows. □

Proof of Lemma 4.10. We first show that

$$\lambda_t(S \cap S(=_t \theta)) \le g_t(S \cap S(=_t \theta) \mid S(>_t \theta))$$

for any $\theta \in [0, 1]$.

If $\theta = 1$, then clearly $\lambda_t(S \cap S(=_t \theta)) = 0 \le g_t(S \cap S(=_t \theta) \mid S(>_t \theta))$. Otherwise, fix $\theta \in [0, 1)$. Let $(U_0, V_0, W_0) = ((S(<_t \theta), S(=_t \theta), S(>_t \theta)), \{(U_k, V_k, W_k)\}_k$ be the sequence of partitions induced by (U_0, V_0, W_0), and $\{T_k\}_k$ the corresponding sequence of leading groups. Recall that by the choice of λ_t, we have for each k,

$$\lambda_t(S \cap T_k) \le g_t(S \cap T_k \mid U_k \cup (V_k \setminus T_k)).$$

Now by submodularity of g_t,

$$\begin{aligned}
\lambda_t(S \cap S(=_t \theta)) &= \sum_k \lambda_t(S \cap T_k) \\
&\le \sum_k g_t(S \cap T_k \mid U_k \cup (V_k \setminus T_k)) \\
&\le \sum_k g_t(S \cap T_k \mid S(>_t \theta) \cup (U_k \cap S(=_t \theta) \cap S)) \\
&= g_t(S \cap S(=_t \theta) \mid S(>_t \theta)).
\end{aligned}$$

Given the above inequality, we have

$$\lambda_t(S) = \sum_{\theta: S(=_t \theta) \neq \emptyset} \lambda_t(S \cap S(=_t \theta)) \le \sum_{\theta: S(=_t \theta) \neq \emptyset} g_t(S \cap S(=_t \theta) \mid S(>_t \theta)) = g_t(S).$$ □

D OMITTED PROOFS IN SECTION 4.6

Proof of Theorem 4.4. We first show that

$$\mathbb{E}[f(S_t)] = \sum_S x_t(S) \cdot f(S).$$

W.l.o.g., assume $q_t(i) \ge q_t(i+1)$ for $i \in [n-1]$, and let $q_t(n+1) = 0$. Then

$$\mathbb{E}[f(S_t)] = \mathbb{E}\left[f(\{i\} \mid [i-1]) \cdot \mathbb{I}[i \in S_t]\right] = \sum_i f(\{i\} \mid [i-1]) \cdot q_t(i).$$

On the other hand,

$$\sum_S x_t(S) \cdot f(S) = \sum_i (q_t(i+1) - q_t(i)) \cdot f([i]) = \sum_i (q_t(i+1) - q_t(i)) \sum_{j \le i} f(\{j\} \mid [j-1])$$

$$= \sum_j f(\{j\} \mid [j-1]) \sum_{i \ge j} (q_t(i+1) - q_t(i)) = \sum_j f(\{j\} \mid [j-1]) \cdot q_t(j) = \mathbb{E}[f(S_t)].$$

Now through a similar argument, one may show that for any $\tau \in [0, t]$,

$$\mathbb{E}[g_\tau([n] \setminus S_\tau)] = \sum_S x_t(S) \cdot g_\tau([n] \setminus S) \Longrightarrow \int_0^t \mathbb{E}[g_\tau([n] \setminus S_\tau)] \, \mathrm{d}\tau = \int_0^t \sum_S x_\tau(S) \cdot g_\tau([n] \setminus S) \, \mathrm{d}\tau.$$

The theorem follows. □

E OMITTED PROOFS IN SECTION 5

Proof of Theorem 5.1. Fix n to be a large enough number, and $\varepsilon > 0$ to be determined later. For $i \in [n]$, let $e_i : 2^{[n]} \to \mathbb{R}_+$ be such that $e_i(S) = \mathbb{I}[i \in S]$. We construct g_t such that for any positive integer k and $t \in [k-1, k)$, $g_t = \alpha \cdot e_i$ for some $i \in [n]$ and $\alpha \ge 0$. Recall that an XOS function is the maximum of a number of additive clauses. We first construct a random hard instance where clauses of f arrive online, which is relatively intuitive and easy to reason about. Then we present a symmetrization argument to turn the construction with online clauses of f into one with online resources, substantiated by moving all randomness to g_t. To be specific, we first make a simplifying assumption, that clauses of f are added online in each phase, and before a clause is added, it is not observable by the algorithm. Under this assuption, we construct f and g_t such that the gap between any algorithm and the offline optimal strategy cannot be bounded by a constant. We will show how to effectively implement this when the algorithm actually has access to the entire f from time 0.

We construct the instance in phases, in an inductive manner. In the first phase, we fix $r_1 = 0$, let $c_1 = e_1$, and $g_t = \infty \cdot e_1$ for $t \in [0, 1)$. We add c_1 to f and finish the first phase. In the k-th phase where $k > 1$, we construct a new clause c_k of f. We flip a biased coin r_k, which is 0 w.p. $1 - \varepsilon$ and 1 w.p. ε. If $r_k = 0$, then the phase consists of 1 resource, and if $r_k = 1$, the phase consists of 2 resources. Let $s_k = \sum_{0 < \ell < k} r_\ell$. Observe that the number of resources appearing in phases before k is exacly $k - 1 + r_k$. For each phase k, let $g_t = \varepsilon \cdot e_{k+s_k}$ for $t \in [k - 1 + s_k, k + s_k)$. That is, the renting cost of the first resource in phase k is ε. If $r_k = 0$, let

$$c_k = e_{k+s_k} + \sum_{\ell < k} r_\ell \cdot e_{\ell + s_\ell}.$$

We add clause c_k to f and finish phase k. Otherwise, let $g_t = \infty \cdot e_{k+s_k+1}$ for $t \in [k + s_k, k + 1 + s_k)$. That is, the renting cost of the second resource in phase k is ∞. In other words, the algorithm has to purchase this resource. Let

$$c_k = e_{k+s_k} + e_{k+s_k+1} + \sum_{\ell < k} r_\ell \cdot e_{\ell + s_\ell}.$$

We then add c_k to f and finish the phase. See Table 1 for an example of the construction.

Now fix an algorithm. We condition on the random bits of the algorithm and consider its behavior on the above random instance. Observe that the instance is constructed, such that after the first phase, the algorithm can purchase one resource "for free" per phase. As a result, the algorithm has to make exactly one decision per phase, i.e., whether to purchase the first (and possibly the only) resource in the phase. In fact, if the algorithm knew beforehand that $r_k = 0$, then the better choice would be to purchase the first (and only) resource in the phase, and the cost in the phase would be 0. Otherwise, the algorithm would be better off by purchasing not the first, but the second resource

resource	**1**	2	**3**	4	**5**	**6**	7	**8**	9	**10**	**11**	**12**
c_1 ($r_1 = 0$)	**1**											
c_2 ($r_2 = 1$)		1	**1**									
c_3 ($r_3 = 1$)		1		1	**1**							
c_4 ($r_4 = 0$)		1		1		**1**						
c_5 ($r_5 = 1$)		1		1			1	**1**				
c_6 ($r_6 = 1$)		1		1			1		1	**1**		
c_7 ($r_7 = 0$)		1		1			1		1		**1**	
c_8 ($r_8 = 0$)		1		1			1		1			**1**
g_t	∞	$\boldsymbol{\varepsilon}$	∞	$\boldsymbol{\varepsilon}$	∞	ε	$\boldsymbol{\varepsilon}$	∞	$\boldsymbol{\varepsilon}$	∞	ε	ε

Table 1. An example realization of the construction. Items purchased in the offline optimal strategy and costs incurred in g_t and each clause c_k are highlighted in boldface. Observe that the cost incurred in each clause is exactly 1, so the max is also 1. The corresponding total cost is $1 + 4\varepsilon$.

in the phase, at a cost of ε for renting the first resource. This is because in such cases, the algorithm has to purchase the second resource to avoid the ∞ renting cost. Then purchasing the first resource costs 1 and renting costs only ε.

Now since the clause is not added to f before the phase ends, the algorithm cannot observe r_k before making the decision. At phase $k > 1$, there are then two cases depending on the choice of the algorithm.

- The algorithm chooses not to purchase the first resource. This is in some sense a safer option, since if $r_k = 1$ and the second resource shows up, then the algorithm can always purchase it at no cost. The cost incurred in the phase is then deterministically ε.
- The algorithm chooses to purchase the first resource. This means the algorithm decides to take some risk. If, with probability $1 - \varepsilon$, $r_k = 0$ and the second resource does not show up, the cost incurred by the algorithm is 0. But if, with probability ε, $r_k = 1$ and the second resource shows up, then the algorithm has to purchase the second resource, paying another 1. The expected cost of the algorithm is therefore again ε.

So, no matter what the algorithm chooses, the expected cost in each phase is always ε, and after K phases, the total expected cost is $1 + K \cdot \varepsilon$.

On the other hand, the expected cost of the offline optimal strategy is much smaller.

- With probability $1 - \varepsilon$, $r_k = 0$, and the offline strategy incurs cost 0 by purchasing the first resource.
- With probability ε, $r_k = 1$, and the offline strategy incurs cost ε by purchasing the second resource.

So, after K phases, the expected cost of the offline optimal strategy is $1 + K \cdot \varepsilon^2$. As $K \longrightarrow \infty$, the ratio between the two costs, $(1 + K \cdot \varepsilon)/(1 + K \cdot \varepsilon^2)$, tends to ε^{-1}, which can be arbitrarily large by taking ε to be small. Note that this crucially depends on n being large too, since in each phase we need at least 1 new resource. Taking expectation over the random bits of the algorithm (or by Yao's Minimax Lemma), this implies a gap of $\omega(1)$ between the algorithm and the offline benchmark.

The above lower bound works under the assumption that the algorithm cannot observe furture clauses not yet added to f. This is not true in our model. For instance, in Table 1, the algorithm can infer that $r_2 = 1$ by only looking at the first 2 columns. This is because the second column has value 1 in each row below c_2. Even if the algorithm has only value oracle access to g_t, it can still obtain information about r_2 by querying $g_1(\{2, 4\})$. We now show how to remove this assumption.

The idea is to symmetrize the construction, and confuse the algorithm by creating all possible future clauses. At the very beginning, we create 2^{k-1} clauses for each phase k,

$$\left\{c_k^{b_1=0, b_2, \ldots, b_k}\right\}_{(b_2, \ldots, b_k) \in 2^{k-1}} .$$

Intuitively, $c_k^{b_1, b_2, \ldots, b_k}$ is the right k-th clause given realization $(b_2, \ldots, b_k)$ of $k-1$ random bits. To single out this clause, we prepare 2^{k-1} resources as candidates of the "first resource" in the k-th clause, one for each candidate clause. We choose the right realization by letting the corresponding resource show up at the beginning of the k-th phase, so the right clause dominates other candidates in terms of the cost. Also we prepare a common candidate of the second reource in each phase, which may or may not effectively show up depending on r_k. Let $\overline{b_1 b_2 \ldots b_k}$ be the integer whose binary representation is "$b_1 b_2 \ldots b_k$". The clauses are then constructed recursively (but all presented at the beginning) as follows. For any $(b_2, \ldots, b_k) \in 2^{k-1}$,

$$c_k^{b_1, \ldots, b_k} = c_{k-1}^{b_1, \ldots, b_{k-1}} - e_{(2^{k-2}-1)+(k-2)+2^{k-2}} + e_{(2^{k-1}-1)+(k-1)+2^{k-1}} + e_{(2^{k-1}-1)+(k-1)+\overline{b_1 \ldots b_k}}.$$

For the k-th phase, we now draw an additional random bit, v_k, which is 0 with probability 0.5 and 1 with probability 0.5. Again we fix $v_1 = 0$. We use v_k as a noise to make the $r_k = 0$ and $r_k = 1$ cases appear symmetric. When the k-th phase begins, we draw r_k and v_k, and let $b_k = v_k \oplus r_k \oplus 1$ be the XOR of v_k, r_k and 1. That is, b_k is v_k if $r_k = 1$, and $(1 - v_k)$ if $r_k = 0$. We demand the $\overline{v_1 v_2 \ldots v_{k-1} b_k}$-th candidate resource in phase k (i.e., resource $(2^{k-1}-1) + (k-1) + \overline{v_1 v_2 \ldots v_{k-1} b_k}$). In other words, for $t \in [k-1+s_k, k+s_k)$, we let $g_t = \varepsilon \cdot e_{(2^{k-1}-1)+(k-1)+\overline{v_1 v_2 \ldots v_{k-1} b_k}}$. Then if $r_k = 0$, we finish the phase. Otherwise, we demand the common candidate for the phase, i.e., resource $(2^{k-1}-1) + (k-1) + 2^{k-1}$. That is, for $t \in [k+s_k, k+s_k+1)$, we let $g_t = \infty \cdot e_{(2^{k-1}-1)+(k-1)+2^{k-1}}$. See Table 2 for the initial part of the construction.

resource	1	2^1-1 $+1+$ $\overline{00}$	2^1-1 $+1+$ $\overline{01}$	2^1-1 $+1+$ 2^1	2^2-1 $+2+$ $\overline{000}$	2^2-1 $+2+$ $\overline{001}$	2^2-1 $+2+$ $\overline{010}$	2^2-1 $+2+$ $\overline{011}$	2^2-1 $+2+$ 2^2
c_1^0	1								
$c_2^{0,0}$		1		1					
$c_2^{0,1}$			1	1					
$c_3^{0,0,0}$		1			1				1
$c_3^{0,0,1}$		1				1			1
$c_3^{0,1,0}$			1				1		1
$c_3^{0,1,1}$			1					1	1
g_t	∞	ε	ε	∞	ε	ε	ε	ε	∞

Table 2. Initial part of the symmetrized construction.

Now observe that the clauses of f are totally symmetric and deterministic. Moreover, the distribution of the first resource demanded in each phase is identical regardless of $\{r_k\}_k$. The algorithm therefore cannot obtain any information by observing the costs and / or the resource presented. Nevertheless, the structure of the extended construction is exactly the same as the intial construction. As a result, the expected cost of the offline optimal strategy after K phases is again $1 + K \cdot \varepsilon^2$, and that of the algorithm is at least $1 + K \cdot \varepsilon$. The gap again goes to ε^{-1}, except that we now need $\Theta(2^K)$ resources to create K phases. Taking $K = \Theta(\log n)$ and $\varepsilon = \omega(1/\sqrt{\log n})$, the above argument implies that no algorithm is $o(\sqrt{\log n})$-competitive. □

Proof of Theorem 5.2. Fix n. Let the cost of purchasing f be such that $f(S) = |S|$, i.e., each resources costs 1. Let $\sigma : [n] \to [n]$ be a uniformly random permutation of $[n]$. As a shorthand, denote $\{\sigma(i) \mid i \in S\}$ by $\sigma(S)$. Consider the following randomized construction of g_t.

$$g_t(S) = \begin{cases} 0, & t \geq n \\ \mathbb{I}[\sigma([n] \setminus [i-1]) \subseteq S] \cdot n^2, & t \in [i-1, i) \text{ where } i \in [n]. \end{cases}$$

In words, $g_t(S)$ is prohibitively large (i.e., n^2) if at time $t \in [i-1, i)$, all resources in $\sigma(\{i, i+1, \ldots, n\})$ are in S, i.e., none of them is purchased.

Fix an algorithm. We condition on the random bits of the algorithm and consider its behavior on the above random instance. Observe the following facts.

- W.l.o.g., the algorithm purchases only at integral $t = i$, when a new constant piece of g_t becomes available to the algorithm.
- At time $t = i - 1$, the algorithm purchases exactly one resource in $\sigma([n] \setminus [i-1])$, if the set purchased right before time t does not contain any resource in $\sigma([n] \setminus [i-1])$. The algorithm makes no purchase otherwise.

Given the above observations, the (cost minimizing) behavior of the algorithm is essentially fixed. We now consider the cost incurred by the algorithm. At time 0, the algorithm purchases some resource in $[n]$. This costs 1. For $i \geq 2$, right before time $t = i - 1$, the intersection between the purchased set and $\sigma([n] \setminus [i-2])$ has size exactly 1. At time $t = i - 1$, two things can happen depending on the realzation of g_t (which given g_{t-1} depends only on $\sigma(i-1)$).

- With probability $1/(n-i+2)$, $\sigma(i-1)$ is the resource in the above intersection. In such cases, the algorithm has to purchase exactly 1 new resource, which costs 1.
- With probability $1 - 1/(n-i+2)$, $\sigma(i-1)$ is not the resource in the intersection. In such cases, the algorithm makes no action till the next integral time.

Overall, the total expected cost of the algorithm is

$$\sum_{2 \leq i \leq n} \frac{1}{n-i+2} = \Omega(\log n).$$

On the other hand, knowing the full realization of σ, the offline optimal strategy is to simply purchase $\sigma(n)$ at time 0, at a total cost of 1. Taking expectation over the random bits of the algorithm (or by Yao's Minimax Lemma), this implies a gap of $\Omega(\log m)$ between the algorithm and the offline benchmark. The lower bound follows. □

Proof of Theorem 5.3. Fix n. Let the purchasing cost f be such that $f(S) = |S|^p$ for some $0 < p < 1$. Let $U = \{u_1, \ldots, u_T\}$ be a uniformly random set of resources of size $T = n^q$ where $0 < q < 1$. Consider the following renting cost g_t.

$$g_t(S) = \begin{cases} 0, & t \geq T \\ \mathbb{I}[u_k \in S] \cdot n^2, & t \in [k-1, k) \text{ where } k \in [T]. \end{cases}$$

In words, $g_t(S)$ is prohibitively large (i.e., n^2) if at time $T > t \in [k-1, k)$, resource u_k is not yet purchased.

Again, fix an algorithm. We condition on the random bits of the algorithm and cosider its behavior on the above random instance. Observe the following facts.

- W.l.o.g., the algorithm purchases only at integral $t = k$, when a new constant piece of g_t becomes available to the algorithm.
- At time $t = k - 1 < T$, the algorithm must purchase resource u_k, if u_k is not already purchased.

- Suppose the total cost of the algorithm is no larger than n^q. Then at any time t, the purchased set has size no larger than n^{pq} (which is the cost of purchasing n^q resources at once).

Now consider gradual realization of the random set U. At time $t = k - 1$, u_k is picked uniformly at random from the $n - (k - 1) \geq n - n^q$ resources not yet picked. On the other hand, the set already purchased right before time t has size at most n^q, which means the probability that u_k is not yet purchased is at least

$$\frac{n - n^q - n^q}{n - n^q} = 1 - \frac{n^q}{n - n^q} = 1 - o(1).$$

So at each time $t = k - 1$, with constant probability the algorithm must purchase something, which costs at least 1. The total cost incurred is therefore at least $\Omega(T) = \Omega(n^q)$. Note that this is based on the assumption that the actual cost of the algorithm does not exceed n^q. So, either the assumption holds, and the total cost is $\Omega(n^q)$, or the assumption does not hold, and the total cost is at least n^q.

On the other hand, knowing U beforehand, the offline optimal strategy is to purchase U at time 0, at a total cost of n^{pq}. This creates a gap of $\Omega(n^{q(1-p)})$, which can be larger than $n^{1-\varepsilon}$ for any $\varepsilon > 0$ by letting $q \longrightarrow 1$ and $p \longrightarrow 0$. Taking expectation over the random bits this implies the desired lower bound. □