JMLR: Workshop and Conference Proceedings vol (2012) 1–19 25th Annual Conference on Learning Theory

Unified Algorithms for Online Learning and Competitive Analysis

Niv Buchbinder NIV.BUCHBINDER@GMAIL.COM
Computer Science Department, Open University of Israel

Shahar Chen SHAHARCH@CS.TECHNION.AC.IL
Department of Computer Science, Technion, Israel

Joseph (Seffi) Naor NAOR@CS.TECHNION.AC.IL
Department of Computer Science, Technion, Israel

Ohad Shamir OHADSH@MICROSOFT.COM
Microsoft Research New England

Editor: Shie Mannor, Nathan Srebro, Bob Williamson

Abstract

Online learning and competitive analysis are two widely studied frameworks for online decision-making settings. Despite the frequent similarity of the problems they study, there are significant differences in their assumptions, goals and techniques, hindering a unified analysis and richer interplay between the two. In this paper, we provide several contributions in this direction. We provide a single unified algorithm which by parameter tuning, interpolates between optimal regret for learning from experts (in online learning) and optimal competitive ratio for the metrical task systems problem (MTS) (in competitive analysis), improving on the results of Blum and Burch (1997). The algorithm also allows us to obtain new regret bounds against "drifting" experts, which might be of independent interest. Moreover, our approach allows us to go beyond experts/MTS, obtaining similar unifying results for structured action sets and "combinatorial experts", whenever the setting has a certain matroid structure.

Keywords: Online Learning, Competitive Analysis, Experts, MTS, Matroids

1. Introduction

Online learning, in its decision-theoretic formulation, captures the problem of a decision-maker who iteratively needs to make decisions in the face of future uncertainty. In each round, the decision-maker picks a certain action from an action set, and then suffers a cost associated with that action. The cost vector is not known in advance, and might even be chosen by an adversary with full knowledge of the decision-maker's strategy. The performance is typically measured in terms of the regret, namely the difference between the total accumulated cost and the cost of an arbitrary *fixed* policy from some comparison class. Non-trivial algorithms usually attain regret which is sublinear in the number of rounds.

While online learning is a powerful and compelling framework, with deep connections to statistical learning, it also has some shortcomings. In particular, it is well-recognized that regret against a *fixed* policy is often too weak, especially when the environment changes over time and thus no single policy is always good. This has led to several papers (e.g., Herbster and Warmuth (1998); Hazan and Seshadhri (2009); Crammer et al. (2010); Rakhlin et al. (2011)) which discuss performance with respect to stronger notions of regret, such as adaptive regret or tracking the best expert. A related shortcoming of online learning is that it does not capture well problems with *states*, where costs depend on the decision-maker's current configuration as well as on past actions. Consider, for instance, the problem of allocating jobs to servers in an online fashion. Clearly, the time it takes to process jobs strongly depends on the system state, such as its overall load, determined by all previous allocation decisions. The notion of regret does not capture this setting well, since it measures the regret with respect to a fixed policy, while assuming that at each step this policy faces the exact same costs.

Thus, one might desire algorithms for a much more ambitious framework, where we need to compete against *arbitrary* policies, including an optimal offline policy which has access to future unknown costs, and where we can model states. Such problems have been intensively studied in the field of *competitive analysis* (for a detailed survey, see Borodin and El-Yaniv (1998)). In such a framework, attaining sublinear regret is hopeless in general. Instead, the main measure used is the *competitive ratio*, that bounds the ratio of the total cost of the decision-maker and the total cost of an optimal offline policy, in a worst-case sense. This usually provides a weaker performance guarantee than online learning, but with respect to a much stronger optimality criterion.

While problems studied under these two frameworks are often rather similar, there has not been much research on general connections between the two. The main reason for this situation (other than social factors stemming from the separate communities studying them) is some crucial differences in the modeling assumptions. For example, in order to model the notion of state, competitive analysis usually assumes a *movement cost* of switching between states. In the online learning framework, this would be equivalent to having an additional cost associated with switching actions between rounds. Another difference is that in competitive analysis one assumes *1-lookahead*, i.e., the decision-maker knows the cost vector in the current round. In contrast, online learning has *0-lookahead*, and the decision-maker does not know the cost vector of the current round until making a decision. Such differences, as stated in Cesa-Bianchi and Lugosi (2006), "have so far prevented the derivation of a general theory allowing a unified analysis of both types of problems" (p. 3).

We note that one particular setting, known as learning from experts (in the online learning framework) and metrical task systems (MTS) with a uniform metric (in the competitive analysis framework), has been jointly studied in Blum and Burch (1997). In particular, the latter paper showed how certain algorithms, based on tuning some parameters, were able to interpolate between a reasonable regret bound and a reasonable competitive ratio. The interpolation was performed using the notion of α-*unfair* competitive ratio, which forces the policy we compete with to pay α times more for the movement cost. In the limit, α goes to infinity, and thus the competing policy becomes essentially static, and the setting becomes reminiscent of online learning.

While these are important and interesting results, they are specific to the setting of experts/MTS. In modern online learning, learning from experts is now known to be a very special case of much more general settings, such as "combinatorial experts" (see Chapter 5 in Cesa-Bianchi and Lugosi (2006)), and online convex optimization. Thus, a natural question is whether unifying analysis and algorithms exist in such cases as well.

Our Contributions: In this paper, we contribute to this research direction by providing a novel unified algorithmic approach, based on recent primal-dual LP techniques developed in competitive analysis (see the survey of Buchbinder and Naor (2009)). First, we show that in the experts/MTS setting, our algorithm attains both optimal regret and an optimal competitive ratio (unlike the results in Blum and Burch (1997), which do not obtain optimal competitive ratios), as well as optimal results for settings in between, such as shifting and drifting experts. The regret bound for drifting experts is new, to the best of our knowledge, and might be of independent interest.

Furthermore, we show how our approach can be applied to more general, "structured" learning/competitive analysis settings, which satisfy *matroid* constraints. Matroids play an important role in combinatorial optimization since the pioneering work of Edmonds in the 1960s and they naturally capture structured action sets such as spanning trees and sparse subsets. In the context of online convex optimization, our results may be viewed as online learning over the matroid base polytope. As in the experts/MTS case, we also get regret bounds against actions which shift or drift a limited amount. Moreover, this can be done in a fine-grained way which respects the problem structure (e.g. competing with spanning trees where only a bounded number of individual edges can change over time). Our algorithms are straightforward, and the various performance guarantees are all obtained just by tuning two parameters.

A key technical feature in our algorithms is that in intermediate steps weights can have negative values, thus deviating from the standard approach of both approximation and online algorithms, and multiplicative updates and weight sharing algorithms.

We emphasize that although some of the settings we discuss might also be treatable by more "conventional" online learning tools, we obtain relevant algorithms naturally from our framework, rather than requiring a case-by-case construction (which is common for online learning over structured sets, see Koolen et al. (2010)).

Overall, we hope that our work on combining online learning and competitive analysis provides a step towards bringing these two rich and mature fields closer together. We also hope that the tools we develop may lead to practical algorithms which combine the advantages of both worlds. On one hand, the practical performance and usefulness of online learning, and on the other hand the robustness to highly dynamic and state-dependent environments of competitive analysis.

Related Work: There are several works related to ours, other than Blum and Burch (1997) which we have already discussed. However, to the best of our knowledge, none of them attempt to provide a single algorithmic approach which connects online learning with competitive analysis. For example, Bansal et al. (2010) show an analysis of experts and the unfair MTS problem, using a primal-dual approach similar to ours. However, a different algorithm and analysis is applied to each of the problems, the algorithms are considerably more complex, and do not scale as well to the more general setting of matroids. Blum et al. (2003) discuss algorithms for decision making on lists and trees, for both a competitive analysis setting and an online learning setting, and show how they can be combined using the hedge algorithm (Freund and Schapire (1997)) to provide simultaneous guarantees. Papers such as Blum et al. (1999) and Abernethy et al. (2010) discuss competitive-analysis algorithms derived using tools from online learning, e.g., regularization. Other works attempt to strengthen the standard regret framework of online learning, such as learning with global cost functions (Even-Dar et al. (2009)) and using more adaptive notions of regret as discussed above. The matroid settings that we consider partially overlap with those of Koolen et al. (2010), which were studied in the standard online learning framework. For these settings, we obtain similar optimal results for online learning, without the need for case-by-case constructions, and again get an interpolation between online learning and competitive analysis.

2. Preliminaries: Online Learning and Competitive Analysis

We begin by describing online learning and competitive analysis, as applied to the settings we consider. To facilitate our unified analysis, we will strive to use the same notation and terminology for both settings, sometimes using conventions from one to describe the other.

Online learning in the experts setting proceeds in T rounds. We consider a finite action set $\mathcal{E}$, where $|\mathcal{E}| = n$. In the beginning of each round t, the decision-maker maintains a distribution vector $\mathbf{w}_{t-1}$ over $\mathcal{E}$ (which can be seen as a randomized policy over picking one out of n "experts" at that round). Then, a *cost vector* $\mathbf{c}_t$ is revealed, and the decision-maker incurs the (expected) cost $\langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle$. Vector $\mathbf{c}_t$ may be generated in an arbitrary, possibly adversarial way, and we only assume that each vector's entry is bounded in $[0, 1]$ (which can be easily relaxed by scaling). The decision-maker then chooses a new vector $\mathbf{w}_t$ for the next round. The goal of the decision-maker is to minimize *regret*, defined as

$$\sum_{t=1}^{T} \langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle - \sum_{t=1}^{T} \langle \mathbf{w}^*, \mathbf{c}_t \rangle,$$

where $\mathbf{w}^* = \arg\min_{\{w \geq 0, \|w\|_1 = 1\}} \{\sum_{t=1}^{T} \langle \mathbf{w}, \mathbf{c}_t \rangle\}$. For this bound to be non-trivial, we expect a regret which grows sublinearly with T. A more ambitious goal studied in the literature (e.g. Herbster and Warmuth (1998)) is tracking the best expert, or regret against "shifting" experts. In that case, we wish to minimize $\sum_{t=1}^{T} \langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle - \sum_{t=1}^{T} \langle \mathbf{w}_t^*, \mathbf{c}_t \rangle$, where $\mathbf{w}_0^*, \ldots, \mathbf{w}_{T-1}^*$ is the best sequence of distributions which change at most k times (i.e. $\mathbf{w}_i^* \neq \mathbf{w}_{i+1}^*$ for at most k values of i). In this paper, we will in fact study a more general framework, which we call "drifting" experts, in which the regret is against the optimal sequence $\mathbf{w}_0^*, \ldots, \mathbf{w}_{T-1}^*$ such that $\sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_1 \leq k$. This generalizes shifting experts, since any k-shifting sequence is also a k-drifting sequence. We are not familiar with existing explicit results in the literature for drifting experts.

In the more general framework that we consider here, rather than just picking single elements of $\mathcal{E}$, we assume that the decision-maker can pick *subsets* of $\mathcal{E}$, from a family of subsets $\mathcal{I}$ which has some structure. Such settings were considered in several online learning papers, such as Kalai and Vempala (2005) and Koolen et al. (2010). For example, consider web advertising, where we can place exactly s ads on some website at any given timepoint, out of n ads overall. This can be naturally modeled as an online learning problem, where $\mathcal{I}$ is all of $\mathcal{E}$'s subsets of size s, and we want to compete against the set of best s ads in hindsight. As another example, consider online learning of spanning trees, which is relevant in the context of communication networks. In that case, $\mathcal{E}$ is a set of edges in a graph, and $\mathcal{C}$ is the convex hull of all subsets of edges which form a spanning tree. The goal in these settings is to minimize regret with respect to the best single element of $\mathbf{w} \in \mathcal{C}$ in hindsight, namely

$$\sum_{t=1}^{T}\langle \mathbf{w}_{t-1}, \mathbf{c}_t\rangle - \min_{\mathbf{w}\in\mathcal{C}}\sum_{t=1}^{T}\langle \mathbf{w}, \mathbf{c}_t\rangle.$$

It turns out that the latter two settings, the basic experts setting, as well as many other settings, satisfy a *matroid* structure. Matroids are extremely useful combinatorial objects[1], which are formally defined as follows, see e.g., Schrijver (2003). Let $\mathcal{E}$ be a finite set and let $\mathcal{I}$ be a nonempty collection of subsets of $\mathcal{E}$, called *independent sets*. $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ is called a matroid if for every $\mathcal{S}_1 \subseteq \mathcal{S}_2$, if $\mathcal{S}_2 \in \mathcal{I}$ then also $\mathcal{S}_1 \in \mathcal{I}$. Additionally, if $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{I}$ and $|\mathcal{S}_1| > |\mathcal{S}_2|$, then there exists an element $e \in \mathcal{S}_1 \setminus \mathcal{S}_2$ such that $\mathcal{S}_2 \cup \{e\} \in \mathcal{I}$. The latter property is called the *set exchange* property. For $\mathcal{S} \subseteq \mathcal{E}$, a subset B of $\mathcal{S}$ is called a *base* of $\mathcal{S}$ if B is a maximal independent subset of $\mathcal{S}$. A well known fact is that for any subset $\mathcal{S}$ of $\mathcal{E}$, any two bases of $\mathcal{S}$ have the same size, called the *rank* of $\mathcal{S}$, denoted by $r(\mathcal{S})$. For example, s-sparse subsets are the bases of an *s-uniform matroid*, where $r(\mathcal{E}) = s$, and experts are the special case with $s = 1$. Spanning trees in a graph $G = (V, E)$ are bases of a *graphic matroid* with $\mathcal{E} = E$ and $\mathcal{I}$ being the collection of all subsets of E that form a forest, with rank $r(\mathcal{E}) = |V| - 1$. The *base polytope* of a matroid $\mathcal{M}$ is defined as the convex hull of the incidence vectors of the bases of $\mathcal{M}$. We refer to this polytope as $B(\mathcal{M})$. The *density* of a matroid $\mathcal{M}$, $\gamma(\mathcal{M})$, is defined as $\max_{\mathcal{S}\subseteq\mathcal{E}, \mathcal{S}\neq\emptyset}\{|\mathcal{S}|/r(\mathcal{S})\}$. For example, the density of the s-subsets matroid is n/s. The density of a graphic matroid (spanning trees) in a graph $G = (V, E)$ is $\max_{S\subseteq V, |S|>1}\{|E(S)|/(|S| - 1)\}$, where $E(S)$ is the set of edges in the subgraph defined by the vertices of S.

We focus on algorithms which work over bases of matroids, interpolating online learning and competitive analysis, and obtaining results in intermediate settings such as competing against shifting and drifting targets. For computational efficiency, our algorithms maintain a distribution $\mathbf{w}_t$ over $\mathcal{E}$ (rather than the possibly-exponentially large $\mathcal{I}$). In competitive analysis, this is known as a *fractional solution*. Since all vertices of $B(\mathcal{M})$ are matroid bases, any such fractional solution always corresponds to a valid distribution over the bases. Hence we may use the fractional solution to actually sample from a consistent distribution on the bases of the matroid. Such a procedure is known as *rounding*. Pipage rounding is an example of a relevant rounding technique which is fast and easy to implement (see Chekuri and Vondrak (2009) for a description). Since these are known techniques, which are not the focus of our paper, we omit the implementation details.

We now turn to describe the matroid general setting in the competitive analysis framework. We first note that the analogue of the experts setting is known as the metrical task system (MTS) problem on a uniform metric, first formulated in Borodin et al. (1992). MTS abstracts many important online decision problems, e.g., process migration. In the online setting, the decision-maker sequentially needs to choose a vector $\mathbf{w}_t$ in a high-dimensional simplex and incur costs depending on arbitrarily-chosen cost vectors. However, there are some important differences.

First, the decision-maker pays a *movement* cost for changing from $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$, which equals $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$, and not only a cost depending on $\mathbf{c}_t$ (known as the *service cost*). Second, the service cost incurred in round t is defined to be $\langle \mathbf{w}_t, \mathbf{c}_t\rangle$, and not $\langle \mathbf{w}_{t-1}, \mathbf{c}_t\rangle$.

1. For instance, they play a crucial role in the analysis of greedy algorithms, and have deep connections to submodular functions which have recently gained popularity in machine learning.

In other words, the decision-maker is allowed to first see the cost vector $\mathbf{c}_t$, and only then choose the new vector $\mathbf{w}_t$ and pay accordingly. This is called *1-lookahead*. In contrast, in the experts setting the decision-maker first pays the cost $\langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle$ and only then chooses a vector $\mathbf{w}_t$. This is called *0-lookahead*. We decompose the total cost paid by the decision-maker into the service cost S_1 (with 1-lookahead) and the movement cost M as follows:

$$S_1 = \sum_{t=1}^{T} \langle \mathbf{w}_t, \mathbf{c}_t \rangle \qquad , \qquad M = \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1.$$

To motivate these notions, we note that in the context of (say) MTS, one thinks of $\mathbf{w}_t$ as a distribution over n possible "states" the algorithm might be in, $\frac{1}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$ as the cost associated with changing that state, and $\mathbf{c}_t$ as specifying the cost of processing a task in each of the n states. Because of the movement cost, the ability of getting the cost $\mathbf{c}_t$ in advance does not trivialize the problem. To allow comparison to the experts setting, we also define $S_0 = \sum_{t=1}^{T} \langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle$ as the service cost of an algorithm whose action at round t does *not* depend on $\mathbf{c}_t$. The framework naturally extends to the context of matroids - the decision-maker needs to maintain over time a base in a matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$.

Another important difference, in comparison to the online learning framework, is the performance measure. In competitive analysis the goal is not to compete against the best fixed element in $B(\mathcal{M})$, but rather against the *optimal offline sequence* $\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*$, which is a solution to

$$\min_{\forall t=1,\ldots,T \ | \ \mathbf{w}_t \in B(\mathcal{M})} \sum_{t=1}^{T} \langle \mathbf{w}_t, \mathbf{c}_t \rangle + \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1.$$

In other words, $\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*$ is the optimal sequence of the decision-maker's choices, had she known all the cost vectors in advance, and could have solved the problem offline. Clearly, this is a much more ambitious goal than minimizing the regret with respect to a fixed $\mathbf{w}^*$. We let

$$S_1^* = \sum_{t=1}^{T} \langle \mathbf{w}_t^*, \mathbf{c}_t \rangle \qquad , \qquad M^* = \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_1$$

denote the service cost and the movement cost of this optimal sequence, and let $\text{OPT} = S_1^* + M^*$ denote the total cost. Thus, the *competitive ratio* is defined as the minimal $c \geq 1$, such that for any sequence of cost vectors,

$$S_1 + M \ \leq \ c \cdot \text{OPT} + d,$$

where d is a constant independent of T. In competitive analysis, c is usually strictly greater than one, and is independent of T. For example, in the MTS setting the attainable competitive ratio is known to be $O(\ln n)$ (Borodin et al. (1992)).

A crucial refinement of competitive ratio, which we use for providing a unified analysis of the two settings, is the notion of *α-unfair* competitive ratio, for $\alpha \geq 1$. This notion modifies the sequence $\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*$ we compete against. Rather than defining it as the sequence minimizing $\sum_{t=1}^{T} \langle \mathbf{w}_t^*, \mathbf{c}_t \rangle + \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_1$, we define it as the sequence which is the solution to:

$$\min_{\forall t=1,\ldots,T \ | \ \mathbf{w}_t \in B(\mathcal{M})} \sum_{t=1}^{T} \langle \mathbf{w}_t, \mathbf{c}_t \rangle + \alpha \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1.$$

The optimal cost of the above is denoted as $\mathrm{OPT}(\alpha)$. In words, the sequence we compete against pays α times more than the decision-maker for movement. The case $\alpha = 1$ corresponds to the standard competitive analysis setting. For $\alpha > 1$, the setting becomes easier, because it encourages the competing sequence to move less. In the limit $\alpha \to \infty$, the optimal sequence necessarily satisfies $\mathbf{w}_1^* = \ldots = \mathbf{w}_T^*$, and the setting becomes reminiscent of online learning where we compare ourselves against a fixed $\mathbf{w}^*$ (although the 1-lookahead and the movement cost features remain). The α-unfair competitive ratio has been proposed in Blum et al. (1992), and used to show connections between online learning and competitive analysis (for experts/MTS) in Blum and Burch (1997).

To facilitate our regret bounds for k-drifting sequences, we let OPT_k denote the cost of the best k-drifting sequence (of valid vectors in $B(\mathcal{M})$) which minimizes $\sum_{t=1}^T \langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle$. It is easy to show that $\mathrm{OPT}(\alpha) \leq \mathrm{OPT}_k + \alpha k$. Another simple observation, based on the boundedness of $\mathbf{c}_t$, is that $S_0 = \sum_{t=1}^T \langle \mathbf{w}_{t-1}, \mathbf{c}_t \rangle \leq \sum_{t=1}^T \langle \mathbf{w}_t, \mathbf{c}_t \rangle + \sum_{t=1}^T \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1 = S_1 + M$. Combining these two, we get the following useful observation which relate the online learning and competitive analysis settings[2]:

Observation 1 *Suppose we have an algorithm (in the α-unfair setting) whose total cost is at most $cOPT(\alpha) + d$, then we have an online learning algorithm with total cost*

$$S_0 \leq S_1 + M \leq cOPT(\alpha) + d \leq cOPT_k + c\alpha k + d. \tag{1}$$

3. Results

We first present our algorithm (Algorithm 1) and results for the experts/MTS setting. We prove the following theorem.

Algorithm 1 Experts/MTS Algorithm (learning-style formulation)

Parameters: $\alpha \geq 1, \eta > 0$
Initialize $w_{i,0} = \frac{1}{n}$ for all $i = 1, \ldots, n$.
for $t = 1, 2, \ldots$ **do**
  Let $(c_{1,t}, \ldots, c_{n,t})$ be the cost vector at time t.
  Using binary search, find the smallest value a_t such that $\sum_{i=1}^n w_{i,t} = 1$, where

$$w_{i,t} = \max\left\{0, \left(w_{i,t-1} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) \mathbf{e}^{-\eta(c_{i,t} - a_t)} - \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right\} \tag{2}$$

end for

Theorem 2 *For any $\alpha \geq 1, \eta > 0$, Algorithm 1 attains*

$$S_1 \ \leq \ OPT(\alpha) + \frac{\ln(n)}{\eta} \quad , \quad M \ \leq \ \left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right) (\eta OPT(\alpha) + \ln(n)). \tag{3}$$

In particular, for $\alpha \to \infty$ (regret against a fixed distribution), by Observation 1, we get

$$S_0 \leq S_1 + M \leq (1 + \eta) OPT(\infty) + \frac{\ln(n)}{\eta} + \ln(n). \tag{4}$$

2. An observation of a similar flavor was also given in Blum and Burch (1997).

By plugging[3] $\alpha = \ln(n)/\eta$ *and using Observation 1, we also obtain*

$$S_0 \leq (1 + 3\eta)\, OPT_k + \frac{(k+1)\ln(n)}{\eta} + 3(k+1)\ln(n). \tag{5}$$

Let us try to understand the bounds in the theorem. For Equation (3), if we set $\alpha = 1$ and $\eta = \ln(n) + \ln\ln n$, we get the best known bound for MTS on uniform metrics (Bansal et al. (2010); Abernethy et al. (2010)). In particular, the bound is better than that obtained by the analysis of Blum and Burch (1997), who also interpolate between experts and MTS. For Equation (4), if we set $\eta = \sqrt{\frac{\ln(n)}{\mathrm{OPT}(\infty)}}$, then our analysis yields a virtually optimal regret bound of $2\sqrt{\mathrm{OPT}(\infty)\ln(n)} + \ln(n)$ for the experts setting. Moreover, it is not hard to see that when $\alpha \to \infty$, our algorithm reduces to the canonical multiplicative updates algorithm (see Cesa-Bianchi and Lugosi (2006)). Equation (5) is a regret bound with respect to the optimal k-drifting sequence. Setting $\eta = \sqrt{(k+1)\ln(n)/3\mathrm{OPT}_k}$, we get an essentially optimal regret of less than $2\sqrt{3(k+1)\ln(n)\mathrm{OPT}_k} + 3(k+1)\ln(n)$ for this problem. We emphasize that while there exist previous results for the case of shifting experts, here we provide an algorithm and analysis for the strictly more general setting of drifting experts[4]. We note that although $\mathrm{OPT}(\infty)$ and OPT_k may not be known in advance in order to tune η, one can use a standard doubling trick to circumvent this (or obtain bounds in which these quantities are replaced by the number of rounds T (Cesa-Bianchi and Lugosi (2006))).

The general case of a matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ is handled by Algorithm 2, which works similarly to Algorithm 1. The algorithm maintains a distribution vector $\mathbf{w}_t \in B(\mathcal{M})$ over the elements of $\mathcal{E}$. Initially, we pick $\mathbf{w}_0$ to be a vector in $B(\mathcal{M})$ such that $\max_{e\in\mathcal{E}}\{\frac{1}{w_{0,e}}\}$ is minimized. By a simple observation there is always such a base such that $\max_{e\in\mathcal{E}}\{\frac{1}{w_{0,e}}\} = \gamma(\mathcal{M})$ and this is the best possible (see Observation 5). In each round, we have an "update" step in which we decrease the value $w_{e,t}$ of each element in the matroid. Note that this can even make the value of $w_{e,t}$ negative. After this step, a sequence of up to n normalization steps is implemented. Before each normalization step we consider the maximal *tight set* with respect to our current solution. A set $\mathcal{S} \subseteq \mathcal{E}$ is tight if $\sum_{e\in\mathcal{S}} w_{e,t} = r(\mathcal{S})$, and it is well known that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are tight, then so are $\mathcal{S}_1 \cap \mathcal{S}_2$ and $\mathcal{S}_1 \cup \mathcal{S}_2$. In particular, there is a maximal tight set which contains all elements whose value $w_{e,t}$ cannot be increased without violating the matroid constraints. In each normalization step we therefore pick all elements which are *not* in a tight set and increase their value, until an additional element joins a tight set. For s-sparse subsets, checking if an element has joined a tight set can be easily done in linear time, and for spanning trees a separation oracle for the forest polytope can be applied (Singh (2008), Theorem 3.8). Generally, the above condition can be checked in polynomial time by a reduction to submodular function minimization (see Schrijver (2003), Chapter 40). The sequence of normalization steps ends when all elements become tight.

The performance guarantee of the algorithm is provided below. We note that aside from a negligible additive factor, it is a natural generalization of Theorem 2, as the expert setting corresponds to a matroid with $r(\mathcal{E}) = 1, \gamma(\mathcal{M}) = n$.

3. This value is chosen for simplicity, and is not the tightest possible.
4. There do exist results for regret against drifting targets in the ℓ_2 norm Zinkevich (2003). However, these results do not require a significant change in the algorithm. In contrast, the standard multiplicative updates algorithm can be shown to fail against ℓ_1 drift, so a new algorithm is indeed required.

Algorithm 2 Matroid Algorithm (learning-style formulation)

Parameters: $\alpha \geq 1, \eta > 0$

Find a fractional base $\mathbf{w}_0 \in B(\mathcal{M})$ such that for each $e \in \mathcal{E}$, $w_{e,0} \geq \frac{1}{\gamma(\mathcal{M})}$.

for $t = 1, 2, \ldots$ **do**

Let $(c_{1,t}, \ldots, c_{n,t})$ be the current cost vector.

(Update step): For each $e \in \mathcal{E}$, let $w_{e,t} = \left(w_{e,t-1} + \frac{1}{\mathbf{e}^{\eta\alpha}-1}\right)\mathbf{e}^{-\eta c_{e,t}} - \frac{1}{\mathbf{e}^{\eta\alpha}-1}$.

(Normalization step): As long as $\sum_{e\in\mathcal{E}} w_{e,t} < r(\mathcal{E})$,

1. Let $\mathcal{S}$ be the set of elements that currently **do not** belong to a tight set.
2. For each $e \in \mathcal{S}$ update $w_{e,t} = \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha}-1}\right)\mathbf{e}^{\eta a_{\mathcal{S},t}} - \frac{1}{\mathbf{e}^{\eta\alpha}-1}$, where $a_{\mathcal{S},t}$ is the smallest value such that there exists $e \in \mathcal{S}$ that joins a tight set.

end for

Theorem 3 *For matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$, and any $\alpha \geq 1, \eta > 0$, Algorithm 2 attains*

$$\begin{aligned} S_1 &\leq OPT(\alpha) + \frac{r(\mathcal{E})}{\eta}\ln\left(\gamma\left(\mathcal{M}\right)\right) + \frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1} \\ M &\leq \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha}-1}\right)\left(\eta OPT(\alpha) + \ln\left(\gamma\left(\mathcal{M}\right)\right)\right) \end{aligned}$$

For $\alpha \to \infty$ (regret against a fixed distribution), by Observation 1, we get

$$S_0 \leq S_1 + M \leq (1+\eta)\, OPT(\infty) + \frac{r(\mathcal{E})\ln\left(\gamma\left(\mathcal{M}\right)\right)}{\eta} + \ln\left(\gamma\left(\mathcal{M}\right)\right). \tag{6}$$

By plugging $\alpha = \ln\left(1 + (n - r(\mathcal{E}) + 1)\ln(n - r(\mathcal{E}) + 1)\right)/\eta$ and using Observation 1,

$$S_0 \leq (1 + 2\eta)\, OPT_k + \frac{2(k + r(\mathcal{E}))\ln(n - r(\mathcal{E}) + 1)}{\eta} + 3(k+1)\ln(n - r(\mathcal{E}) + 1). \tag{7}$$

For Equation (6), if we set $\eta = \sqrt{\frac{r(\mathcal{E})\ln(\gamma(\mathcal{M}))}{\mathrm{OPT}(\infty)}}$, then our analysis yields a regret bound of $2\sqrt{r(\mathcal{E})\ln(\gamma\left(\mathcal{M}\right))\mathrm{OPT}(\infty)} + \ln(\gamma\left(\mathcal{M}\right))$, for the experts setting. For example, for s-sparse subsets, this corresponds to $O\left(\sqrt{s\ln(n/s)\mathrm{OPT}(\infty)} + \ln(n/s)\right)$, and for spanning trees over $|E|$ edges and $|V|$ vertices, we get $O(\sqrt{|V|\ln(|E| - |V| + 1)\mathrm{OPT}(\infty)}) + \ln(|E| - |V| + 1)$. This corresponds to the results of Koolen et al. (2010), and moreover, our latter result is for spanning trees over general graphs rather than complete graphs. Equation (7) provides a version for k-drifting sequence. Setting $\eta = \sqrt{(k + r(\mathcal{E}))\ln(n - r(\mathcal{E}) + 1)/\mathrm{OPT}_k}$, we get regret of less than $4\sqrt{(k + r(\mathcal{E}))\ln(n - r(\mathcal{E}) + 1)\mathrm{OPT}_k} + 3(k+1)\ln(n - r(\mathcal{E}) + 1)$ for this problem. Since the drift is measured with respect to the ℓ_1 norm over $B(\mathcal{M})$, it naturally captures the structure of the problem. In particular, the drift is measured with respect to changes in individual elements in the s-subsets or individual edges in the spanning trees.

(P)	$\min \sum_{t=1}^{T} \sum_{i=1}^{n} c_{i,t} \cdot w_{i,t} + \sum_{t=1}^{T} \sum_{i=1}^{n} \alpha \cdot z_{i,t}$		(D)	$\max \sum_{t=0}^{T} a_t$
$\forall t \geq 0$	$\sum_{i=1}^{n} w_{i,t} = 1$		$\forall i$ and $t = 0$	$a_0 + b_{i,1} \leq 0$
$\forall t \geq 1$ and expert i	$z_{i,t} \geq w_{i,t} - w_{i,t-1}$		$\forall t \geq 1$ and i	$b_{i,t+1} \leq b_{i,t} + c_{i,t} - a_t$
$\forall t$ and expert i	$z_{i,t}, w_{i,t} \geq 0$		$\forall t \geq 1$ and i	$0 \leq b_{i,t} \leq \alpha$

Figure 1: The primal and dual LP formulations for the MTS problem.

4. Proofs and Algorithm Derivation

In this section, we explain how we derive and analyze our algorithms. We focus on the simpler case of experts/MTS (Algorithm 1 and Theorem 2). The derivation in the matroid case is conceptually similar but technically more complex, and is provided in appendix A.

The derivation is based on a primal-dual linear programming analysis. It starts from a very simple LP formulation (Figure 1) of the optimal (offline) α-unfair solution. Note that in order to charge for $\alpha \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$, it suffices to charge only on increasing coordinates. Thus, we will charge both the optimal solution and our algorithm for increasing variables. Figure 1 also contains a description of the dual program (D). This program plays a central role in our analysis. We define D as the value of the dual program. It is well known that D is a lower bound on the value of any primal solution.

To analyze Algorithm 1, it will be more convenient to describe it in the following equivalent form (the equivalence is not hard to show). This form has a more explicit primal-dual structure, and is the standard form used in the competitive analysis community. This form explicitly contains the dual variables of (D).

Algorithm 3 Experts/MTS Algorithm (fractional primal-dual formulation)

Parameters: $\alpha \geq 1, \eta > 0$

Initialize $w_{i,0} = \frac{1}{n}$, $b_{i,1} = \alpha - \frac{\ln\left(\frac{e^{\eta\alpha} + n - 1}{n}\right)}{\eta}$ for all $i = 1, \ldots, n$.

During execution, maintain the relation $w_{i,t} = \max\left\{0, \frac{e^{\eta(\alpha - b_{i,t+1})}}{e^{\eta\alpha} - 1} - \frac{1}{e^{\eta\alpha} - 1}\right\}$

for $t = 1, 2, \ldots$ **do**

  Let $(c_{1,t}, \ldots, c_{n,t})$ be the cost vector at time t.

  (Update step): Set $b_{i,t+1} = b_{i,t} + c_{i,t}$.

  (Normalization step): Using binary search, find the smallest value a_t, and set $b_{i,t+1} = b_{i,t+1} - a_t$, such that $\sum_{i=1}^{n} w_{i,t} = 1$.

end for

We interpret our algorithm as a primal-dual algorithm that increases dual variables and sets the primal variables accordingly. We then show that the dual solution constructed by the algorithm is feasible, and that the cost of our primal solution is bounded by the dual. This will eventually lead to Equation (3) in Theorem 2. The other bounds in the theorem are simple corollaries obtained by a direct calculation.

First, without loss of generality, we can assume that at the end of the normalization step $b_{i,t+1} \leq \alpha$. Simple calculations show that for $b_{i,t+1} = \alpha$, $w_{i,t} = 0$. Thus, when $b_{i,t+1} > \alpha$, then $w_{i,t+1} < 0$ and is therefore set to 0 by the algorithm. If this happens we can run the algorithm with $c'_{i,t} < c_{i,t}$, the smallest value for which $w_{i,t} = 0$. The algorithm's behavior is unchanged (and so is its cost). However, the optimal value of the primal (and so the dual that we compare to) only reduces by decreasing the value of $c_{i,t}$ to $c'_{i,t}$.

We next interpret the normalization step as increasing the value of a_t continuously and setting the dual variable $b_{i,t+1} = b_{i,t} + c_{i,t} - a_t$. In the following, we analyze the performance using a primal-dual method.

Primal (P) is feasible: Clearly, in the beginning $w_{i,0}$ is a feasible solution. By definition we have $w_{i,t} \geq 0$ by the end of each iteration. In addition, $\sum_{i=1}^n w_{i,t} = 1$, which implies $w_{i,t} \leq 1$.

Dual (D) is feasible: Since initially $w_{i,0} = \frac{1}{n}$, then we can set for each i, $b_{i,1} = \alpha - \frac{\ln(\frac{\mathbf{e}^{\eta\alpha}+n-1}{n})}{\eta}$, $a_0 = -\alpha + \frac{\ln(\frac{\mathbf{e}^{\eta\alpha}+n-1}{n})}{\eta}$, and we have that the first dual constraint is feasible. The primal solution is feasible, thus $0 \leq w_{i,t} \leq 1$. By the primal dual relation we get: $0 \leq \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1})}-1}{\mathbf{e}^{\eta\alpha}-1} \leq 1$. Simplifying, we get $0 \leq b_{i,t+1} \leq \alpha$. Finally, the algorithm always keeps the dual constraints with equality: $b_{i,t+1} = b_{i,t} + c_{i,t} - a_t$.

Primal-dual relation: Let ΔD_t be the change in the cost of the dual solution at time t. We bound the cost of the algorithm in each iteration by the change in the cost of the dual.

Bounding the movement cost at time t: Let M_t be the movement cost at time t. As we said we charge our algorithm (and OPT(α)) only for increasing the fractional value of the elements. We get,

$$\begin{aligned} M_t &= \sum_{i=1}^n \max\{0, w_{i,t} - w_{i,t-1}\} = \sum_{i=1}^n \max\left\{0, \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1} - \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1} + c_{i,t} - a_t)} - 1}{\mathbf{e}^{\eta\alpha} - 1}\right\} \\ &\leq \sum_{i=1}^n \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1} - \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1} - a_t)} - 1}{\mathbf{e}^{\eta\alpha} - 1} = \sum_{i=1}^n \left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\left(1 - \mathbf{e}^{-\eta a_t}\right) \\ &\leq \sum_{i=1}^n \left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\eta a_t = \eta\left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right)\Delta D_t, \end{aligned} \tag{8}$$

where Inequality (8) follows since for any x, $\mathbf{e}^x - 1 \geq x$. Thus,

$$\begin{aligned} M &\leq \sum_{t=1}^T \eta\left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right)\Delta D_t \leq \eta\left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right)\left(D + \alpha - \frac{\ln(\frac{\mathbf{e}^{\eta\alpha}+n-1}{n})}{\eta}\right) \\ &\leq \eta\left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right)D + \left(1 + \frac{n}{\mathbf{e}^{\eta\alpha} - 1}\right)\ln(n). \end{aligned}$$

Bounding the service cost: Since the solution is feasible at times t and $t-1$, we get:

$$\begin{aligned} 0 &= \sum_{i=1}^n (w_{i,t-1} - w_{i,t}) = \sum_{i=1}^n \left(\frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1} + c_{i,t} - a_t)} - 1}{\mathbf{e}^{\eta\alpha} - 1} - \frac{\mathbf{e}^{\eta(\alpha - b_{i,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \\ &= \sum_{i=1}^n \left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\left(\mathbf{e}^{\eta(c_{i,t} - a_t)} - 1\right) \geq \sum_{i=1}^n \left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\eta\left(c_{i,t} - a_t\right), \end{aligned} \tag{9}$$

where Inequality (9) follows since for any x, $\mathbf{e}^x - 1 \geq x$. Rearranging we get:

$$\sum_{i=1}^{n}\left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) c_{i,t} \leq \sum_{i=1}^{n}\left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) a_t. \tag{10}$$

Note that,

$$\begin{aligned}
0 &= \sum_{i=1}^{n}(w_{i,T} - w_{i,0}) = \sum_{i=1}^{n}\left(\frac{\mathbf{e}^{\eta(\alpha - b_{i,1} - \sum_{t=1}^{T} c_{i,t} + \sum_{t=1}^{T} a_t)} - 1}{\mathbf{e}^{\eta\alpha} - 1} - \frac{\mathbf{e}^{\eta(\alpha - b_{i,1})} - 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \\
&= \sum_{i=1}^{n}\left(w_{i,0} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\left(\mathbf{e}^{\eta\left(\sum_{t=1}^{T} a_t - \sum_{t=1}^{T} c_{i,t}\right)} - 1\right) \\
&\geq \sum_{i=1}^{n}\left(w_{i,0} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\eta\left(\sum_{t=1}^{T} a_t - \sum_{t=1}^{T} c_{i,t}\right) \qquad (11) \\
&= \eta\left(\frac{1}{n} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right)\left(\sum_{t=1}^{T}\sum_{i=1}^{n} a_t - \sum_{t=1}^{T}\sum_{i=1}^{n} c_{i,t}\right),
\end{aligned}$$

where Inequality (11) follows since $\mathbf{e}^x - 1 \geq x$ for any x. This implies,

$$\sum_{t=1}^{T}\sum_{i=1}^{n} a_t \leq \sum_{t=1}^{T}\sum_{i=1}^{n} c_{i,t}. \tag{12}$$

We can now bound the service cost:

$$\begin{aligned}
\sum_{t=1}^{T}\langle \mathbf{w}_t, \mathbf{c}_t\rangle &\leq \sum_{t=1}^{T}\sum_{i=1}^{n} c_{i,t}\left(w_{i,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) - \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\sum_{t=1}^{T}\sum_{i=1}^{n} a_t \qquad (13) \\
&\leq \sum_{t=1}^{T}\sum_{i=1}^{n} a_t w_{i,t} = \sum_{t=1}^{T} a_t \qquad (14) \\
&= D + \left(\alpha - \frac{\ln(\frac{\mathbf{e}^{\eta\alpha} + n - 1}{n})}{\eta}\right) \leq D + \frac{\ln(n)}{\eta},
\end{aligned}$$

where Inequality (13) follows by Inequality (12), Inequality (14) follows by Inequality (10).

Acknowledgments

Niv Buchbinder was supported in part by ISF grant 954/11 and BSF grant 2010426. Seffi Naor was supported in part by the Google Inter-university center for Electronic Markets and Auctions, by ISF grant 954/11, and by BSF grant 2010426.

References

J. Abernethy, P. L. Bartlett, N. Buchbinder, and I. Stanton. A regularization approach to metrical task systems. In *ALT*, 2010.

N. Bansal, N. Buchbinder, and J. Naor. Towards the randomized k-server conjecture: A primal-dual approach. In *SODA*, pages 40–55, 2010.

A. Blum and C. Burch. On-line learning and the metrical task system problem. In *COLT*, 1997.

A. Blum, H. J. Karloff, Y. Rabani, and M. E. Saks. A decomposition theorem and bounds for randomized server problems. In *FOCS*, 1992.

A. Blum, C. Burch, and A. Kalai. Finely-competitive paging. In *FOCS*, 1999.

A. Blum, S. Chawla, and A. Kalai. Static optimality and dynamic search-optimality in lists and trees. *Algorithmica*, 36(3):249–260, 2003.

A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

A. Borodin, N. Linial, and M. E. Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, 1992.

N. Buchbinder and J. Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2009.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

C. Chekuri and J. Vondrak. Randomized pipage rounding for matroid polytopes and applications. *CoRR*, abs/0909.4348, 2009.

K. Crammer, Y. Mansour, E. Even-Dar, and J. Wortman Vaughan. Regret minimization with concept drift. In *COLT*, 2010.

E. Even-Dar, R. Kleinberg, S. Mannor, and Y. Mansour. Online learning for global cost functions. In *COLT*, 2009.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *ICML*, 2009.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998.

A. T. Kalai and S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

W. M. Koolen, M. K. Warmuth, and J. Kivinen. Hedging structured concepts. In *COLT*, 2010.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Beyond regret. In *COLT*, 2011.

A. Schrijver. *Combinatorial Optimization: polyhedra and efficientcy.* Springer, 2003.

M. Singh. *Iterative Methods in Combinatorial Optimization.* PhD thesis, Tepper School of Business, Carnegie Mellon University, 2008.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

(P)	$\min \sum_{t=1}^{T} \sum_{e\in\mathcal{E}} c_{e,t} \cdot w_{e,t} + \alpha \cdot \sum_{t=1}^{T} \sum_{e\in\mathcal{E}} z_{e,t}$	(D)	$\max \sum_{t=0}^{T} \left(r(\mathcal{E})a_t - \sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S}) a_{\mathcal{S},t} \right)$
$\forall t \geq 0$ and $\mathcal{S}\subset\mathcal{E}$	$\sum_{e\in\mathcal{S}} w_{e,t} \leq r(\mathcal{S})$	$\forall e\in\mathcal{E}$	$a_0 - \sum_{\mathcal{S}\mid e\in\mathcal{S}} a_{\mathcal{S},0} + b_{e,1} \leq 0$
$\forall t \geq 0$	$\sum_{e\in\mathcal{E}} w_{e,t} = r(\mathcal{E})$	$\forall t\geq 1$ and $e\in\mathcal{E}$	$b_{e,t+1} \leq b_{e,t} + c_{e,t} - a_t + \sum_{\mathcal{S}\mid e\in\mathcal{S}} a_{\mathcal{S},t}$
$\forall t \geq 1$ and $e\in\mathcal{E}$	$z_{e,t} \geq w_{e,t} - w_{e,t-1}$	$\forall t\geq 1$ and $e\in\mathcal{E}$	$b_{e,t} \leq \alpha$
$\forall t$ and $e\in\mathcal{E}$	$z_{e,t}, w_{e,t} \geq 0$	$\forall t, e\in\mathcal{E}, \mathcal{S}\subset\mathcal{E}$	$b_{e,t}, a_{\mathcal{S},t} \geq 0$

Figure 2: The primal and dual LP formulations for the Matroid problem.

Appendix A. Proofs and Algorithm Derivation - the Matroid Case

In this Section we analyze Algorithm 2 that works for the general matroid setting, and prove Theorem 3. As in the case of experts/MTS (uniform matroid), it is more convenient to analyze our algorithm in an equivalent form which has an explicit primal-dual structure.

Algorithm 4 Matroid Algorithm (fractional primal-dual formulation)

Parameters: $\alpha \geq 1, \eta > 0$.

During the execution of the algorithm maintain the relation: $w_{e,t} = f(b_{e,t+1}) = \frac{\mathbf{e}^{\eta(\alpha - b_{e,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1}$.

Find a fractional base in the matroid such that for each e: $w_{e,0} \geq \frac{1}{\gamma(\mathcal{M})}$, and set $b_{e,1}$ accordingly.

for $t = 1, 2, \ldots$ **do**

Let $(c_{1,t}, \ldots, c_{n,t})$ be the current cost vector.

(Update step): Set $b_{e,t+1} = b_{e,t} + c_{e,t}$.

(Normalization step): As long as $\sum_{e\in\mathcal{E}} w_{e,t} < r(\mathcal{E})$:

1. Let $\mathcal{S}$ be the set of elements that currently **do not** belong to a tight set.
2. For each $e \in \mathcal{S}$ update $b_{e,t+1} \leftarrow b_{e,t+1} - a_{\mathcal{S},t}$, where $a_{\mathcal{S},t}$ is the smallest value such that there exists $e\in\mathcal{S}$ that joins a tight set.

end for

For our analysis, we need the following properties of matroids.

Claim 4 *In any matroid for any set $\mathcal{S} \subseteq \mathcal{E}$:*

$$\frac{|\mathcal{S}|}{n - r(\mathcal{E}) + 1} \leq r(\mathcal{S}). \tag{15}$$

If $r(\mathcal{E}\setminus\mathcal{S}) < r(\mathcal{E})$ then:

$$\frac{|\mathcal{S}|}{n - r(\mathcal{E}) + 1} \leq r(\mathcal{E}) - r(\mathcal{E}\setminus\mathcal{S}). \tag{16}$$

Proof If $r(\mathcal{E}\setminus\mathcal{S}) = r(\mathcal{E})$,

$$\frac{|\mathcal{S}|}{n - r(\mathcal{E}) + 1} = \frac{|\mathcal{S}|}{(|\mathcal{S}| + |\mathcal{E} \setminus \mathcal{S}|) - r(\mathcal{E} \setminus \mathcal{S}) + 1} \leq \frac{|\mathcal{S}|}{|\mathcal{S}| + 1} < 1 \leq r(\mathcal{S}).$$

If $r(\mathcal{E} \setminus \mathcal{S}) < r(\mathcal{E})$,

$$\begin{aligned} \frac{|\mathcal{S}|}{n - r(\mathcal{E}) + 1} &= \frac{|\mathcal{S}|}{|\mathcal{S}| + 1 - (r(\mathcal{E}) - |\mathcal{E} \setminus \mathcal{S}|)} \leq \frac{|\mathcal{S}|}{|\mathcal{S}| + 1 - (r(\mathcal{E}) - r(\mathcal{E} \setminus \mathcal{S}))} \\ &\leq r(\mathcal{E}) - r(\mathcal{E} \setminus \mathcal{S}) \leq r(\mathcal{S}), \qquad (17) \end{aligned}$$

where Inequality (17) follows as $(k-1)/(k-x) \leq x$ for any $1 \leq x \leq k-1$. Inequality (16) is proved using a similar argument. ■

Observation 5 *There is always a fractional base in the matroid such that for each e:* $w_{e,0} \geq \frac{1}{\gamma(\mathcal{M})}$.

Proof It is known that any fractional solution in the matroid polytope can be extended (by only increasing variables) to a matroid base. Thus, we only need to prove that $w_{e,0} = \frac{1}{\gamma(\mathcal{M})}$ is in the matroid polytope. Hence, we should prove that for any $\mathcal{S} \subseteq \mathcal{E}$: $\sum_{e \in \mathcal{S}} w_{e,0} = \frac{|\mathcal{S}|}{\gamma(\mathcal{M})} \leq r(\mathcal{S})$, which follows by the definition of $\gamma(\mathcal{M})$. ■

We now turn to derive Theorem 3. As in the case of Theorem 2, we focus on proving Inequality (6), as the other bounds in the theorem follow by simple calculations.

We interpret line (2) in the algorithm as increasing the value of a_t and $a_{\mathcal{S}}$ for elements in a tight set continuously and setting the dual variable $b_{e,t+1} = b_{e,t} + c_{e,t} - a_t + \sum_{\mathcal{S}|e \in \mathcal{S}} a_{\mathcal{S}}$. In the following, we analyze the performance using primal-dual method.

Primal (P) is feasible: By Observation 5, $w_{e,0}$ is a feasible solution. By induction on the steps, we prove that the algorithm produces a feasible solution (and $\mathbf{w}_t$ remains in the domain $B(\mathcal{M})$) . The update step reduces the value of each $w_{e,t}$. Then, in the normalization step the value of each $w_{e,t}$ can only grow. Since there are at most n elements, after at most n iterations $\mathcal{E}$ is tight and thus the solution is feasible. Note that the algorithm never increases elements in tight sets. Finally, by the end of the normalization step we get for all e, $w_{e,t} \geq 0$, otherwise if $w_{e,t} < 0$ then $\sum_{e' \in N \setminus \{e\}} w_{e',t} > r(\mathcal{E}) \geq r(N \setminus \{e\})$ which violates the matroid constraints.

Dual is feasible: Since initially, $w_{e,0} \geq \frac{1}{\gamma(\mathcal{M})}$, then we may set for each e, $b_{e,1} \leq \alpha - \frac{\ln(\frac{\mathbf{e}^{\eta\alpha} + \gamma(\mathcal{M}) - 1}{\gamma(\mathcal{M})})}{\eta}$. Thus, by setting $a_0 = -\alpha + \frac{\ln(\frac{\mathbf{e}^{\eta\alpha} + \gamma(\mathcal{M}) - 1}{\gamma(\mathcal{M})})}{\eta}$ and setting $a_{\mathcal{S},0} = 0$ for all $\mathcal{S} \subset \mathcal{E}$, we have that the first set of dual constraints is feasible. The primal solution is feasible, thus $0 \leq w_{e,t} \leq 1$. By the primal dual relation we get: $0 \leq \frac{\mathbf{e}^{\eta(\alpha - b_{e,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1} \leq 1$. Simplifying we get $0 \leq b_{e,t+1} \leq \alpha$. Finally, by the algorithm construction we always keep the dual constraints with equality: $b_{e,t+1} = b_{e,t} + c_{e,t} - a_t + \sum_{\mathcal{S}|e \in \mathcal{S}} a_{\mathcal{S}}$.

Primal-dual relation: Let ΔD be the change in the cost of the dual solution. We bound the cost of the algorithm in each iteration by the change in the cost of the dual.

Bounding the movement cost at time t: Let M_t be the movement cost at time t. We charge our algorithm (and OPT(r)) only for increasing the fractional value of the elements. We get,

$$\begin{aligned}
M_t &= \sum_{e\in\mathcal{E}} \max\{0, w_{e,t} - w_{e,t-1}\} \\
&= \sum_{e\in\mathcal{E}} \max\left\{0, \frac{\mathbf{e}^{\eta(\alpha-b_{e,t+1})}-1}{\mathbf{e}^{\eta\alpha}-1} - \frac{\mathbf{e}^{\eta\left(\alpha-b_{e,t+1}+c_{e,t}-a_t+\sum_{\mathcal{S}:e\in\mathcal{S}} a_{\mathcal{S},t}\right)}-1}{\mathbf{e}^{\eta\alpha}-1}\right\} \\
&\le \sum_{e\in\mathcal{E}} \frac{\mathbf{e}^{\eta(\alpha-b_{e,t+1})}-1}{\mathbf{e}^{\eta\alpha}-1} - \frac{\mathbf{e}^{\eta\left(\alpha-b_{e,t+1}-a_t+\sum_{\mathcal{S}:e\in\mathcal{S}} a_{\mathcal{S},t}\right)}-1}{\mathbf{e}^{\eta\alpha}-1} \\
&= \sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha}-1}\right)\left(1-\mathbf{e}^{-\eta\left(a_t-\sum_{\mathcal{S}:e\in\mathcal{S}} a_{\mathcal{S},t}\right)}\right) \\
&\le \sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha}-1}\right)\eta\left(a_t-\sum_{\mathcal{S}:e\in\mathcal{S}} a_{\mathcal{S},t}\right) && (18)\\
&= \eta\left(r(\mathcal{E})\,a_t - \sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S})\,a_{\mathcal{S},t} + \frac{n\cdot a_t - \sum_{\mathcal{S}\subset\mathcal{E}}|\mathcal{S}|\cdot a_{\mathcal{S},t}}{\mathbf{e}^{\eta\alpha}-1}\right) && (19)\\
&\le \eta\left(1+\frac{n-r(\mathcal{E})+1}{\mathbf{e}^{\eta\alpha}-1}\right)\Delta D_t, && (20)
\end{aligned}$$

where Inequality (18) follows since for any x, $\mathbf{e}^x - 1 \ge x$. Inequality (19) follows since variable $a_{\mathcal{S},t}$ is nonzero only if $\mathcal{S}$ is a tight subset. Inequality (20) follows since,

$$\frac{n\cdot a_t - \sum_{\mathcal{S}\subset\mathcal{E}}|\mathcal{S}|\cdot a_{\mathcal{S},t}}{n-r(\mathcal{E})+1} \le r(\mathcal{E})a_t - \sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S})a_{\mathcal{S},t},$$

which follows as,

$$\begin{aligned}
\frac{n}{n-r(\mathcal{E})+1}a_t - \sum_{\mathcal{S}\subset\mathcal{E}}\frac{|\mathcal{S}|}{n-r(\mathcal{E})+1}a_{\mathcal{S},t} &= \frac{n}{n-r(\mathcal{E})+1}\left(a_t - \sum_{\mathcal{S}\subset\mathcal{E}} a_{\mathcal{S},t}\right) + \sum_{\mathcal{S}\subset\mathcal{E}}\frac{|\mathcal{E}\setminus\mathcal{S}|}{n-r(\mathcal{E})+1}a_{\mathcal{S},t} \\
&\le r(\mathcal{E})\left(a_t - \sum_{\mathcal{S}\subset\mathcal{E}} a_{\mathcal{S},t}\right) + \sum_{\mathcal{S}\subset\mathcal{E}}\left(r(\mathcal{E})-r(\mathcal{S})\right)a_{\mathcal{S},t} && (21)\\
&= r(\mathcal{E})a_t - \sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S})a_{\mathcal{S},t},
\end{aligned}$$

where Inequality (21) is implied by Claim 4, and noticing that $a_t \geq \sum_{\mathcal{S}\subset\mathcal{E}} a_{\mathcal{S},t}$ for any $0 \leq t \leq T$ as, at any moment, the algorithm raises at most one variable $a_{\mathcal{S},t}$ along with a_t. Thus,

$$\begin{aligned}
M &\leq \sum_{t=1}^{T} \eta \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \Delta D_t \\
&\leq \eta \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \left(D + \alpha - \frac{\ln\left(\frac{\mathbf{e}^{\eta\alpha} + \gamma(\mathcal{M}) - 1}{\gamma(\mathcal{M})}\right)}{\eta}\right) \\
&= \eta \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) D + \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \ln\left(\frac{\gamma(\mathcal{M}) \cdot \mathbf{e}^{\eta\alpha}}{\mathbf{e}^{\eta\alpha} + \gamma(\mathcal{M}) - 1}\right) \\
&\leq \eta \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) D + \left(1 + \frac{n - r(\mathcal{E}) + 1}{\mathbf{e}^{\eta\alpha} - 1}\right) \ln(\gamma(\mathcal{M}))
\end{aligned}$$

Bounding the service cost: First note that similarly to the uniform case we have:

$$\begin{aligned}
0 &= \sum_{e\in\mathcal{E}} (w_{e,t-1} - w_{e,t}) = \sum_{e\in\mathcal{E}} \left(\frac{\mathbf{e}^{\eta\left(\alpha - b_{e,t+1} + c_{e,t} - \left(a_t - \sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right)\right)} - 1}{\mathbf{e}^{\eta\alpha} - 1} - \frac{\mathbf{e}^{\eta(\alpha - b_{e,t+1})} - 1}{\mathbf{e}^{\eta\alpha} - 1} \right) \\
&= \sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) \left(\mathbf{e}^{\eta\left(c_{e,t} - \left(a_t - \sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right)\right)} - 1\right) \\
&\geq \sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) \eta \left(c_{e,t} - \left(a_t - \sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right)\right), \qquad (22)
\end{aligned}$$

where Inequality (22) follows since for any x, $\mathbf{e}^x - 1 \geq x$. Rearranging we get:

$$\sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) c_{e,t} \leq \sum_{e\in\mathcal{E}} \left(w_{e,t} + \frac{1}{\mathbf{e}^{\eta\alpha} - 1}\right) \left(a_t - \sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right) \qquad (23)$$

In addition, since $0 \leq b_{e,t} \leq \alpha$ for all e, t then,

$$\sum_{t=1}^{T} \left(a_t - \sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right) = \sum_{t=1}^{T} c_{e,t} + b_{e,1} - b_{e,T} \leq \sum_{t=1}^{T} c_{e,t} + \alpha \qquad (24)$$

Now we can bound the service cost:

$$\begin{aligned}
\sum_{t=1}^{T}\langle \mathbf{w}_t, \mathbf{c}_t\rangle &\leq \sum_{t=1}^{T}\sum_{e\in\mathcal{E}} c_{e,t}\left(w_{e,t}+\frac{1}{\mathbf{e}^{\eta\alpha}-1}\right)-\frac{1}{\mathbf{e}^{\eta\alpha}-1}\sum_{t=1}^{T}\sum_{e\in\mathcal{E}}\left(a_t-\sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right)+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1} \quad (25)\\
&\leq \sum_{t=1}^{T}\sum_{e\in\mathcal{E}} w_{e,t}\left(a_t-\sum_{\mathcal{S}|e\in\mathcal{S}} a_{\mathcal{S},t}\right)+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1} \quad (26)\\
&= \sum_{t=1}^{T}\left(r(\mathcal{E})\,a_t-\sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S})\,a_{\mathcal{S},t}\right)+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1}\\
&= D-\left(r(\mathcal{E})\,a_0-\sum_{\mathcal{S}\subset\mathcal{E}} r(\mathcal{S})\,a_{\mathcal{S},0}\right)+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1}\\
&= D+\frac{r(\mathcal{E})}{\eta}\ln\left(\frac{\gamma(\mathcal{M})\cdot\mathbf{e}^{\eta\alpha}}{\mathbf{e}^{\eta\alpha}+\gamma(\mathcal{M})-1}\right)+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1}\\
&\leq D+\frac{r(\mathcal{E})}{\eta}\ln(\gamma(\mathcal{M}))+\frac{n\alpha}{\mathbf{e}^{\eta\alpha}-1},
\end{aligned}$$

where Inequality (25) follows by Inequality (24), Inequality (26) follows by Inequality (23).