Online Linear Programming with Unobserved Constraints

Wenzhuo Yang yangwenzhuo08@gmail.com
National University of Singapore
Shie Mannor shie@ee.technion.ac.il
Technion – Israel Institute of Technology
Huan Xu huan.xu@isye.gatech.edu
Georgia Institute of Technology

Editor: Anonymous

Abstract

We consider online linear programming with unobserved constraints (LPUC) – a generalization of stochastic linear optimization – where in each round a learner chooses a solution and subsequently receives some feedback about the *feasibility* of the selected solution w.r.t. the unknown constraints, e.g., indicating which constraint is violated or how much the solution deviates from the feasibility set. To tackle this problem, we develop two algorithms, namely, LPUC-ED based on the epsilon-decreasing strategy and LPUC-UCB based on the upper confidence bound strategy, and derive finite time bounds on the regret and the constraint violation.

Keywords: Online Learning, Linear Programming, Optimization

1. Introduction

In this paper, we tackle linear programming with unknown constraints (LPUC) from a dynamic perspective: the decision maker can make a tentative decision, collect feedback information about the decision, and fine tune the decision, essentially solving the LP problem via *trial and error*. We motivate our setup using the following example. Network flow problems, often used to model traffic in a road system and packet flow through network, etc., can be formulated as LP problems. The decision maker who aims to find the maximum flow or the minimum-cost flow does not always know the capacities or costs of all the edges in the network exactly. This paper aims to develop methods to leverage such post-decision information to obtain near optimal solutions in a learning fashion.

To gather the information about the unknown constraints, we consider an online setting where the decision maker or learner selects a solution in each round and then receives corresponding feedbacks providing information about the feasibility of the selected solution. As an example, consider that routers forward data packets through a data network and observe packet delays due to congestion (i..e, flows exceed the edge capacities). The goal is to find solutions close to the optimal solution of the unknown LP. This model generalizes both stochastic linear optimization Dani et al. (2008); Rusmevichientong and Tsitsiklis (2008) and multi-armed bandit problems Lai (1987); Cesa-Bianchi and Lugosi (2006); Pandey et al. (2007); Mannor and Shamir (2011); Abe and Long (1999); Auer et al. (2002); Filippi et al.

(2010); Chu et al. (2011); Abbasi-Yadkori et al. (2011), allowing to tackle a broad class of problems.

To tackle this problem, we develop two algorithms – LPUC-ED based on the epsilon-decreasing strategy Kuleshov and Precup (2000) and LPUC-UCB based on the upper confidence bound strategy Auer et al. (2002); Audibert et al. (2009); Filippi et al. (2010). We measure their performance using *two metrics simultaneously*, namely, *regret* – the difference between the learner's cumulated cost and the cost of the optimal strategy, and *constraint violation* – an indicator of level of constraint violation over the T rounds. We show that the regret and constraint violation of LPUC-ED are $O(dT^{\frac{2}{3}}\log T)$ and $O(dT^{\frac{2}{3}})$ respectively, whereas those of LPUC-UCB are both $O(d\sqrt{T}\log T)$. LPUC-UCB achieves a better regret than LPUC-ED and matches the lower bound of the linear bandit problem Dani et al. (2008) up to a logarithmic factor, but is computationally more demanding than LPUC-ED.

Notations: We use boldface lower-case letters to represent column vectors and capital letters for matrices, and use $[c]_+$ to denote $\max\{0, c\}$. We use $\mathbf{e}_1, \cdots, \mathbf{e}_d$ to represent the standard basis of $\mathbb{R}^d$ and define $\mathbf{e}_{d+1} \triangleq \mathbf{0}$ for convenience, and use $\mathcal{S}_{d-1}(B)$ to denote the unit sphere $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = B\}$.

2. Problem Setting

More generally, we consider to solve a sequence of linear programming problems $\{\mathcal{P}_1, \cdots, \mathcal{P}_T\}$. For each t, $\mathcal{P}_t$ has the following form:

$$\min\ \mathbf{c}^\top \mathbf{x},\ \text{s.t.}\ \mathbf{A}^\top \mathbf{x} \leq \mathbf{b},\ \mathbf{x} \in \mathcal{S}_t, \tag{1}$$

where $\mathcal{S}_t$ is a bounded convex set and $\mathbf{A} \in \mathbb{R}^{d\times m}, \mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^d$ are shared for all t. This paper concerns to find its optimal solution in the case that $\mathbf{A}$ and $\mathbf{b}$ are unknown but $\mathbf{c}$ is known, and assumes that for any input $\mathbf{x} \in \mathcal{S}$ the system provides us *some feedback* about how much $\mathbf{x}$ deviates from the feasibility set, e.g., indicating which constraint is violated. Let $\mathbf{x}^*(t)$ be the optimal solution of $\mathcal{P}_t$. We tackle this problem in the following online setting. In each round t, the learner receives linear program $\mathcal{P}_t$ and chooses a solution $\mathbf{x}(t)$ for $\mathcal{P}_t$. After $\mathbf{x}(t)$ is submitted, she receives the corresponding feedback $\mathbf{r}(t)$ whose i^{th} entry $r_i(t) = f(\mathbf{a}_i^\top \mathbf{x}(t) - b_i) + \xi_i(t)$, where $\mathbf{a}_i$ is the ith column of $\mathbf{A}$, $\xi_i(t)$ is a random noise with mean $\mathbf{0}$ and $f(\cdot)$ is a non-decreasing function. Without loss of generality, we assume that $f(0) = 0$. The goal of the learner is to find the optimal solution of $\mathcal{P}_t$ as t grows. If the cost vector $\mathbf{c}$ in Problem (1) is also unknown, one can convert Problem (1) into its epigraph form. Thus, the problem studied in this paper is more general than linear bandit problems.

Our model is more challenging than linear bandit and generalized linear bandit models because 1) it has more than one unknown constraints so that the preference of a solution can not be simply represented by one scalar; 2) it requires to satisfy multiple objectives simultaneously – the select solutions should be approximately "optimal" (with a nearly optimal objective value) and "feasible" (not far away from the feasible set specified by the unknown constraints) at the same time, while linear bandit and generalized linear bandit models have only one objective – maximizing the cumulated rewards, and therefore it can not be reformulated as a linear bandit problem.

We now discuss necessary assumptions on decision sets $\mathcal{S}_t$, noise $\xi_i(t)$ and function $f(\cdot)$:

1. For any $t = 1, \cdots, T$, Problem (1) is always feasible, i.e., $\mathcal{S}_t \cap \{\mathbf{x} : \mathbf{A}^\top \mathbf{x} \leq \mathbf{b}\} \neq \emptyset$, and there exists constants L and B so that $\|\mathbf{x}\|_2 \leq L$ for any $\mathbf{x} \in \mathcal{S}_t$ and $[-B, B]^d \subseteq \mathcal{S}_t$.

2. The function $f(\cdot)$ is continuously differentiable, Lipschitz continuous with constant l_f, and satisfies $c_f \triangleq \inf_{\mathbf{x} \in \bigcup_{t=1}^T \mathcal{S}_t, (\mathbf{a},b) \in \bigcup_{i=1}^m \mathcal{A}_i} \frac{df(z)}{dz}\Big|_{z=\mathbf{a}^\top \mathbf{x} - b} > 0$. Here $\mathcal{A}_i$ represents the admissible sets for $\mathbf{a}_i$ and b_i, i.e., $(\mathbf{a}_i, b_i) \in \mathcal{A}_i$.

3. For all $t \geq 1$, random variable $\xi_i(t)$ has support $[-R, R]$ and satisfies that $\mathbb{E}[\xi_i(t) | \boldsymbol{\xi}(t-1), \cdots, \boldsymbol{\xi}(1), \mathbf{x}(t), \cdots, \mathbf{x}(1)] = 0$ almost surely.

4. The constraint $\mathbf{A}^\top \mathbf{x} \leq \mathbf{b},\ \mathbf{x} \in \mathcal{S}_t$ is regular, i.e., $\mathbf{b}$ is an interior point of $\{\mathbf{A}^\top \mathbf{x} + \mathbf{z} : \mathbf{x} \in \mathcal{S}_t, \mathbf{z} \in \mathbb{R}^m_+\}$ where $\mathbb{R}^m_+$ denotes the non-negative orthant in $\mathbb{R}^m$.

The desirable solutions should be approximately feasible and optimal at the same time. To measure "optimality" and "feasibility", we consider the following *absolute regret* (or *regret* for short) and *constraint violation*:

$$\text{Regret}(T) = \sum_{t=1}^{T} |\mathbf{c}^\top \mathbf{x}(t) - \mathbf{c}^\top \mathbf{x}^*(t)|,\ \text{Violation}(T) = \sum_{i=1}^{m} \sum_{t=1}^{T} [\mathbf{a}_i^\top \mathbf{x}(t) - b_i]_+.$$

Our notion of regret is different from the traditional regret that sums $\mathbf{c}^\top \mathbf{x}(t) - \mathbf{c}^\top \mathbf{x}^*(t)$, for the following reason. Due to the existence of the unknown constraint, $\mathbf{c}^\top \mathbf{x}(t)$ can be much less than $\mathbf{c}^\top \mathbf{x}^*(t)$ if an infeasible $\mathbf{x}(t)$ is chosen, which makes the traditional regret meaningless as the sum contains both positive and negative terms. A careful reader may notice that this regret also penalizes solutions with smaller objective values than the optimal value due to the absolute-difference loss function, which forces the learner to choose solutions that are close to the optimal one. Overall, our aim is to design policies with both the regret and constraint violation growing sub-linearly in T.

3. Algorithms

Algorithm 1 is developed based on the epsilon-decreasing strategy (LPUC-ED). In the t^{th} round, the first step of Algorithm 1 is estimating $\mathbf{A}$ and $\mathbf{b}$ based on the historical information $\{\mathbf{x}(1), \cdots, \mathbf{x}(t-1), \mathbf{r}(1), \cdots, \mathbf{r}(t-1)\}$ obtained before this round. For convenience, we let $\mathbf{y}(t) = (\mathbf{x}(t)^\top, -1)^\top$ and define several useful quantities:

$$\mathbf{M}_t \triangleq \sum_{k=1}^{t-1} \mathbf{y}(k)\mathbf{y}(k)^\top,\ g_t(\mathbf{z}) \triangleq \sum_{k=1}^{t-1} f(\mathbf{z}^\top \mathbf{y}(k))\mathbf{y}(k),\ \mathbf{g}_t^i \triangleq \sum_{k=1}^{t-1} r_i(k)\mathbf{y}(k)\ \forall i = 1, \cdots, m. \quad (2)$$

Suppose that the admissible set for $(\mathbf{a}_i, b_i)$ is known and denoted by $\mathcal{A}_i$. The new estimates of $\mathbf{a}_i$ and b_i can be computed via the following optimization problem:

$$\mathbf{a}_i(t), b_i(t) = \arg \min_{(\mathbf{a},b) \in \mathcal{A}_i} \|g_t((\mathbf{a}^\top, b)^\top) - \mathbf{g}_t^i\|_{\mathbf{M}_t^{-1}}. \quad (3)$$

As discussed in Filippi et al. (2010), this problem can be easily solved via Newton's method. The second step is selecting $\mathbf{x}(t)$ by solving Problem (1) with the current estimates of $\mathbf{A}$

Algorithm 1 LP with unobserved constraints via the epsilon-decreasing strategy (LPUC-ED)

Input: Cost vector $\mathbf{c} \in \mathbf{R}^d$, decision sets $\mathcal{S}_t$.
Procedure:
1) Play $\mathbf{e}_1, \cdots, \mathbf{e}_{d+1}$ and receive $\mathbf{r}(1), \cdots, \mathbf{r}(d+1)$;
for $t = d+2$ to T **do**
2) Compute $\mathbf{M}_t$ and $\mathbf{g}_t^i$ for $i \in [m]$ via Eqn (2);
3) Compute $\mathbf{a}_i(t), b_i(t)$ for $i \in [m]$ via solving (3);
4) Compute the optimal solution $\hat{\mathbf{x}}(t)$ of the following linear program:

$$\begin{array}{ll} \min & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{a}_i(t)^\top \mathbf{x} \leq b_i(t), \ \forall i = 1, \cdots, m, \\ & \mathbf{x} \in \mathcal{S}_t. \end{array} \tag{4}$$

5) Set variable $\eta(t)$ to $\eta(t) = \begin{cases} \tilde{\eta}(t), & \text{Problem (4) is feasible,} \\ 1, & \text{otherwise} \end{cases}$ where $\tilde{\eta}(t)$ is drawn from Bernoulli distribution with success probability $p(t) \propto 1/t^{1/3}$;
6) Play $\mathbf{x}(t) = [1 - \eta(t)]\hat{\mathbf{x}}(t) + \eta(t)\tilde{\mathbf{x}}(t)$ and receive $\mathbf{r}(t)$, where $\tilde{\mathbf{x}}(t)$ follows the uniform distribution on $\mathcal{S}_{d-1}(B)$;
end for

and $\mathbf{b}$ as shown in (4). We then sample $\mathbf{x}(t)$ from the uniform distribution on $\mathcal{S}_{d-1}(B)$ to explore more information about $\mathbf{A}$ and $\mathbf{b}$ when either Problem (4) is infeasible or $\tilde{\eta}(t)$ – a Bernoulli random variable with parameter $p(t)$ – equals 1. We prove in the next section that the regret and the constraint violation for Algorithm 1 are at most $O(dT^{\frac{2}{3}} \log T)$ and $O(dT^{\frac{2}{3}})$, respectively.

To achieve a regret bound better than $O(dT^{\frac{2}{3}} \log T)$, we develop Algorithm 2 – Linear programming with unobserved constraints via UCB (LPUC-UCB) – that chooses $\mathbf{x}(t)$ by solving a non-convex optimization problem as shown in (5). As opposed to Algorithm 1, this scheme automatically balances exploration and exploitation, and does not require "pure exploration" step. In the next section, we will show that the regret bound and the constraint violation of Algorithm 2 are $O(d\sqrt{T} \log T)$. We remark that in general, it is difficult to obtain the global optimal solution of Problem (5) due to the non-convexity of its constraints. However, we want to highlight that our main concern is the *sample complexity*, i.e., making minimal number of trials, rather than the computation complexity. Moreover, When $f(\cdot)$ is convex, Problem (5) is a DC (difference of convex functions) programming problem that can be solved by many DC algorithms An and Tao (2005) proposed in recent years. When $\mathcal{S}_t$ is a finite discrete set, (5) can be solved by evaluating each element in $\mathcal{S}_t$, i.e., selecting the one that is feasible and has the smallest objective value.

4. Performance Guarantees

This section analyzes the performance of Algorithms 1 and 2. Recall $\mathcal{S}_t$ are bounded convex set, under which Problem (1) is a standard linear programming problem. Theorem 1 and

Algorithm 2 LP with unobserved constraints via UCB (LPUC-UCB)

Input: Cost vector $\mathbf{c} \in \mathbf{R}^d$, decision sets $\mathcal{S}_t$ and parameter $\theta(t)$.
Procedure:
1) Play $\mathbf{e}_1, \cdots, \mathbf{e}_{d+1}$ and receive $\mathbf{r}(1), \cdots, \mathbf{r}(d+1)$;
for $t = d+2$ to T **do**
2) Compute $\mathbf{M}_t$ and $\mathbf{g}_t^i$ for $i \in [m]$ via Eqn(2);
3) Compute $\mathbf{a}_i(t), b_i(t)$ for $i \in [m]$ via solving (3);
4) Solve the optimization problem:

$$\begin{array}{ll} \min & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & f(\mathbf{a}_i(t)^\top \mathbf{x} - b_i(t)) \le \theta(t) \|\mathbf{y}\|_{\mathbf{M}_t^{-1}}, \ \forall i = 1, \cdots, m, \\ & \mathbf{y} = (\mathbf{x}^\top, -1)^\top, \ \mathbf{x} \in \mathcal{S}_t, \end{array} \tag{5}$$

and denote the optimal solution by $\hat{\mathbf{x}}(t)$;
5) Play $\mathbf{x}(t) = \hat{\mathbf{x}}(t)$ and receive $\mathbf{r}(t)$;
end for

Theorem 2 show the upper bounds for the regret and the constraint violation of Algorithms 1 and 2, respectively.

Theorem 1 *Under Assumptions 1–4, there exist constants c, c_1, c_2, c_3 so that for $0 < \delta < 1$, when*

$$T \ge T_0 \triangleq c_2 \left(\frac{l_f R d}{c_f^2} \sqrt{m \log \frac{m^{3/2}}{c_f \delta}} \right)^3, \ \textit{and } \theta(t) \triangleq \frac{c_1 l_f R}{c_f} \sqrt{d \log \frac{2m(L^2+1)t}{\delta d B}},$$

with probability at least $1 - 2\delta - cT^{-9}$ the regret and the constraint violation of Algorithm 1 satisfy

$$\begin{aligned} \textit{Regret}(T) &\le 2T_0 L \|\mathbf{c}\|_2 + c_3 \left[\frac{\theta(T)\sqrt{md}}{c_f} + L\|\mathbf{c}\|_2 \right] T^{2/3} \sqrt{\log T}, \\ \textit{Violation}(T) &\le (T_0 + c_3 T^{2/3}) \sum_{i=1}^m (L\|\mathbf{a}_i\|_2 + |b_i|) + \frac{10 m \theta(T)}{c_f} \sqrt{dT \log T}. \end{aligned}$$

Theorem 2 *Under Assumptions 1–4, there exist constants c_1, c_2 so that for $0 < \delta < 1$, when*

$$\theta(t) = \frac{c_1 l_f R}{c_f} \sqrt{d \log \frac{2m(L^2+1)t}{\delta d B}}, \ \textit{and } T > d+1, \tag{6}$$

with probability at least $1 - 2\delta$ the regret and the constraint violation of Algorithm 2 satisfy

$$\begin{aligned} \textit{Regret}(T) &\le 2(d+1) L \|\mathbf{c}\|_2 + \frac{c_2 \theta(T) \|\mathbf{c}\|_2}{c_f} \sqrt{dT \log T}, \\ \textit{Violation}(T) &\le (d+1) \sum_{i=1}^m (L\|\mathbf{a}_i\|_2 + |b_i|) + \frac{20 m \theta(T)}{c_f} \sqrt{dT \log T}. \end{aligned} \tag{7}$$

Theorem 2 asserts that the regret and the constraint violation of Algorithm 2 are at most $O(d\sqrt{T}\log T)$. Dani et al. (2008) proved that the regret for the linear bandit problem with arbitrary convex compact decision sets has a $\Omega(d\sqrt{T})$ lower bound. Since our model is a general form of the linear bandit problem, the upper bounds achieved by Algorithm 2 are near optimal, i.e., they match the lower bound up to a logarithmic factor.

5. Experiments

This section investigates the empirical performance of our algorithms on synthetic data. The linear programming problems are randomly generated as follows: 1) cost vector $\mathbf{c}$ is sampled from $[-1,1]^d$ uniformly at random, 2) $\mathbf{b}$ is uniformly drawn from $[0,2]^m$, 3) each column of $\mathbf{A}$ is sampled from $\mathcal{S}_{d-1}(1)$ uniformly at random, and 4) $\boldsymbol{\xi}(t)$ is set to $0.01\boldsymbol{\mu}$ where $\boldsymbol{\mu}$ follows the standard Gaussian distribution $\mathcal{N}(0,\mathbf{I})$. We let $\mathcal{A}_i$ – the admissible set for $\mathbf{a}_i$ and b_i – be $[-5,5]^{d+1}$ and $\mathcal{S}_t$ – the decision set in round t – be $[-5,5]^d$. We repeat each test 10 times and report the average results. Problem (3) is solved via the L-BFGS-B algorithm Byrd et al. (1995). For LPUC-ED, $p(t)$ and B are set to $0.1/t^3$ and 5, respectively. For LPUC-UCB, the non-convex optimization problem (5) is solved by two steps: 1) we compute $\tilde{\mathbf{x}}_t$ – the optimal solution of (5) with $\theta(t)=0$ (if $\tilde{\mathbf{x}}_t$ does not exist, $\tilde{\mathbf{x}}_t$ is set to $\mathbf{x}_{t-1}$), and 2) by taking $\tilde{\mathbf{x}}_t$ as the initial solution, we use the SciPy optimization package Jones et al. (2001–) to solve (5) with $\theta(t)=0.1\sqrt{\log t}$. In the first experiment, the linear programming problem is generated with $d=10$ and $m=20$. We compare the acceleration versions of LPUC-ED and LPUC-UCB with parameter $\gamma=0.01$. The empirical performance is measured by three quantities: the regret, the constraint violation and the estimation error. The estimation error is the difference between the true optimal solution $\mathbf{x}^*$ of (1) and the average of the solutions up to time T, namely, $\|\mathbf{x}^*-\frac{1}{T}\sum_{i=1}^{T}\mathbf{x}(T)\|_2$. For input $\mathbf{x}(t)$, we consider two different feedbacks: 1) linear feedback $r_i(t)=\mathbf{a}_i^\top\mathbf{x}(t)-b_i+\xi_i(t)$, and 2) sign feedback $r_i(t)=-1$ if $\mathbf{a}_i^\top\mathbf{x}(t)-b_i\le 0$ and 1 otherwise. In the algorithms, we use $f(x)=(\exp(x)-1)/(\exp(x)+1)$ to approximate the sign function. Figures 1(a) and 1(b) show the empirical performance of LPUC-ED and LPUC-UCB. We observe that their regrets and constraint violations are sub-linear in T, and LPUC-UCB has a significantly better performance than LPUC-ED.

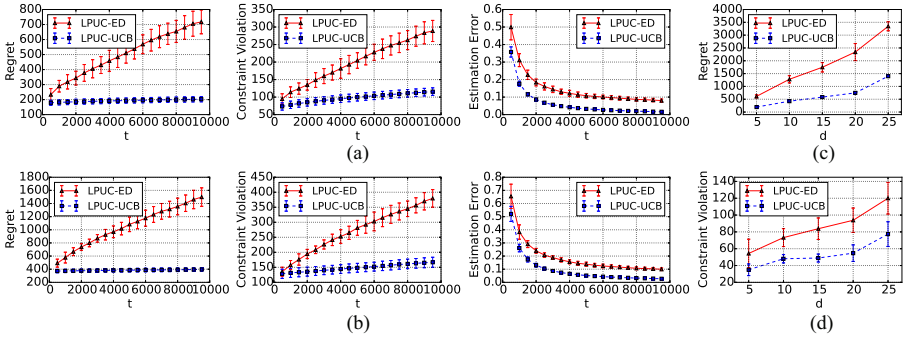

Figure 1: We compare the empirical performance of LPUC-ED and LPUC-UCB. (a) Linear feedback. (b) Sign feedback. (c)(d) The regret and the constraint violation against dimension d with linear feedbacks.

References

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, 2011.

N. Abe and P. M. Long. Associative reinforcement learning using linear probabilistic concepts. In *ICML*, 1999.

L. Hoai An and P. D. Tao. The DC programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133:23–46, 2005.

J. Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. *Journal of Machine Learning Research*, 15:208–214, 2011.

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, 2008.

S. Filippi, O. Capp, A. Garivier, and C. Szepesvari. Parametric bandits: The generalized linear case. In *NIPS*, 2010.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`.

V. Kuleshov and D. Precup. Algorithms for the multi-armed bandit problem. *Journal of Machine Learning Research*, 1:1–48, 2000.

L. T. Lai. Adaptive treatment allocation and the multi-armed bandit problem. *Annals of Statistics*, 15(3):1091–1114, 1987.

S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *NIPS*, 2011.

S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. In *ICML*, 2007.

P. Rusmevichientong and J. N. Tsitsiklis. Linearly parameterized bandits. *CoRR*, 2008.