Online Linear Programming with Uncertain Constraints

Lin Yang
Department of
Infromation Engineering
Chinese University of Hong Kong
yanglin.cuhk@gmail.com

Mohammad H. Hajiesmaili
College of Information
and Computer Sciences
University of Massachusetts Amherst
hajiesmaili@cs.umass.edu

Wing S. Wong
Department of
Infromation Engineering
Chinese University of Hong Kong
wswong@ie.cuhk.edu.hk

***Abstract*—There are many applications scenarios in different disciplines where the critical knowledge of decision making arrives in a sequential manner, so the optimization must be done in an online fashion. An important class of online optimization problems that have been extensively studied in the past is online linear programs. This paper tackles a general class of online linear programs that take into account the online arrival of the constraint entries related to the available budget and demand for packing and covering problems. This generalization is motivated by many recent applications on revenue management or resource allocation problems with the unknown and time-varying budget. As the main contribution of this paper, we propose a decoupling strategy that can be used to reduce the general problem into a series of subproblems with constant constraint entries. Using the proposed strategy, one can decouple the general problem, leverage the state-of-the-art algorithms for the online subproblems with fixed constraint entries, and achieve the same performance for the general problem. As for a case study, we apply the strategy to an extension of the one-way trading problem with the dynamic budget.**

I. Introduction

In many optimization communities such as economics, operations research, and computer science, there is a category of problems in which the complete knowledge of the inputs to the problem is not available in advance. In other words, the inputs to the problem arrive sequentially in an online fashion. These scenarios are known as *online optimization or decision making under uncertainty* that has been studied using a wide range of theoretical tools and in a broad set of applications. In online optimization, the decisions are made in an online fashion, while the uncertainty of future inputs makes it usually impossible to achieve the optimal solution. Hence, there has been a substantial effort to propose online algorithms using different tools and performance metrics. Notable examples are competitive algorithms with the competitive ratio as the performance metric [1]–[3], online learning with the regret as the performance metric [4], [5], and reinforcement learning with different modeling techniques such as Markov decision process and designing optimal policies in expectation [6]. Recent notable application scenarios in which theoretical results have had an impact for real-world design include data center optimization [7]–[9], energy systems [10]–[14], cloud management [15], [16], computer and communication networks [17]–[20], and beyond.

This paper studies a category of online linear programming problems with uncertain constraints using the online competitive framework and competitive ratio as the performance metric. In what follows a brief research background along with related works is presented.

A. Background and Related Works

In many applications on revenue management and resource allocation [21]–[24], the underlying problem could be modeled as a linear program, where the coefficients of the objective function arrive online. Generally, these problems can be either formulated as a minimization or maximization versions on online linear program. The offline formulation of online linear program is as follows.

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} p(i)x(i) \\ \text{subject to} & \sum_{i=1}^{n} a(i,k)x(i) \le b(k), \;\; k = 1,2,\ldots,m. \\ \text{variables} & x(i) \in \mathbb{R}^{+}. \end{array} \tag{1}$$

One can interpret Problem (1) as maximizing the profit given a limited set of resources. More specifically, the objective, which is the sum of a linear combinations of variable $x(i)$ weighted by coefficient $p(i)$, can be considered as the cumulative profit obtained across n time slots. Then, entries $b(k)$, defines the budget/capacity constraint on $x(i)$, and could be interpreted as the limitation of physical resources, e.g., storage capacity in [23], [24]. In the terminology of this paper, $b(k)$ is called the *budget* entry for the resource of type k, $p(i)$ that determines the earned profit by consuming a unit of resource, is called the *price* entry, and $a(i,k)$ that determines the amount of resource consumption is called the *weight* entry.

Problem (1) can be solved in offline manner, however, many applications emphasize the need for online solution design, in which all or a subset of the entries can be only accessed piece by piece as time goes on. For example, in the one-way trading problem [25], as a special case of problem (1), the price entries $p(i)$ arrive online. In online packing problem [26], [27], price and weight entries arrive online.

For the aforementioned problems, there are extensive works that develop *online competitive algorithms* for the underlying online problems that make decisions at each time slot, knowing the past and current online inputs, but not knowing those same inputs for the future. The classic approach for analysis of online algorithms is competitive the analysis [1], where the goal is to design algorithms with the smallest *competitive ratio*. Competitive ratio of an online algorithm is defined as the maximum possible profit ratio between an *offline* optimal algorithm that has access to *complete* input sequence and the online algorithm. The goal is to devise online algorithms that have provably the smallest competitive ratio.

Using the competitive framework, there is an extensive literature both one-way and packing problems and for the state-of-the-art algorithms and detailed competitive analysis of these problems, we refer to [25]–[27].

Besides the aforementioned *maximization* problems, there are several online cost *minimization* problems that can be seen as a converse version of the maximization problems. In these problems, instead of having packing constraints and budget entries, one deal with covering constraints and *demand* entries. For example, in the online covering optimization problem [26], [27], the demand entries are known and given but the weight entries $[a(i,j)]_{j=1}^{m}$ are revealed to the algorithm one by one in an online fashion. An special case is the k-min search problem [28] that minimizes the linear objective of $\sum_{i=1}^{n} c(i)x(i)$, in which the entries $c(i)$ represents the cost and subject to the constraint $\sum_{i=1}^{n} x(i) \geq k$. The k-min search problem can be explained by an application scenario where one wants to buy at least k items from a time-varying market using the minimum payment. At the ending time, the remaining items must be bought if the number of the items is less than k. In the k-min search problem, the demand entry, i.e., k, as the target number of items to be purchased, is given in advance, however, the objective (cost entries) is revealed online. More details about k-min search problem along with optimal online algorithms for this problem is studied in [28].

B. Problem Statement

In the previous literature on online linear packing/covering problems, either the price/cost, weight, or both arrive online. *This paper studies the case that in addition to the above entities, the budget/demand entries also arrive in online manner.*

The interpretation of online arrival of budget/demand entries using the previously mentioned examples is as follows. In one-way trading, the extension considers dynamic changes in the availability of the budget over time. In k-min search, it captures the case that the number of items to be bought is not fixed and will change as time goes on.

This extension is motivated by several recent application scenarios. A notable example is the online offering problem of storage-assisted renewable energy in the deregulated electricity market [24]. The decision maker is a renewable generation company who wants to sell renewable power to the market with a goal of maximizing the profit. In addition, the renewable plant is equipped by on-site energy storage systems that can be used to store renewable generation for future use. The price for the admitted power is dynamic and is settled based on dynamics to balance between supply and demand. The profit maximization problem for the renewable company can be represented as an extension of one-way trading problem with renewal of budget at each slot [24], in which both the market price and budget (renewable generation) entries arrive in online manner. The question is, heavily dependent on the environmental conditions, the energy harvest of renewable plants from the environment is rather random, and it is very challenging for a renewable company to accurately predict its output. With uncertainties in both price and renewable generation, we have a more complicated online linear program that can not be modeled by the existing algorithms for one-way trading algorithm.

In the same way, the online cost minimization problem can be also extended to a more general version where both the objective (price entries) and the demand entries arrive online. For example, in a data center, an significant portion of workload is time-shiftable with a deadline. Upon arrival of a new workload, the execution of workload must be scheduled within the deadline. In reality, the cost of the data center on processing one unit of the demand and the amount of demand can be both revealed online. This results in a cost minimization problem with online arrival of cost and demand entries, extending the k-min search problem.

The cost minimization problem as the k-min search problem and the online covering problem can be formulated in a similar manner as the maximization problems. In this paper, we only formulate the problem maximization versions. It is worth noting that the our proposed decoupling strategy can be extended to the minimization problems as well. With online price and resource entries, we reformulate the general online linear program as follows.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} p(i)x(i) \\ \text{subject to} \quad & \sum_{i=1}^{j} a(i,k)x(i) \leq b(j,k), \\ & j = 1,2,\ldots,n, \;\; k = 1,2,\ldots,m. \\ \text{variables} \quad & x(i) \in \mathbb{R}^{+}. \end{aligned} \tag{2}$$

In the above problem, all the entries $p(i)$, the weight entries $[a(i,k)]_{k=1}^{m}$, and the budget entries $b(j,k)$ are time-varying and arrive online. To the best of our knowledge, this setting with online arrival of all entries to the problem is not studied in the literature, and all the existing work tackle online linear problems with a subset of entries as the online input [25]–[29].

C. Our Contribution

In this paper, we focus on maximization version of online linear problems and propose a novel decoupling strategy for the online linear programs with online budget entries. Using the proposed strategy, the original problem can be decoupled to multiple simpler subproblems each of which with a fixed set of *offline* budget entries. In this way, one can map the subproblems into a well-studied class of problems, and guarantee the same performance, i.e., competitive ratio, for the general case with unknown budget entries. In other word, our result demonstrates that with extreme uncertainty in *all* entries, the same level of performance guarantee could be achieved as compared to the case with *partial* uncertainty in the input. While our results is presented for the maximization problem, the strategy could be extended for the minimization problems as well.

As for a case study, we apply the decoupling strategy to solve the one-way trading problem [25] with online arrival of budget entries over time. We show that by using our strategy, we can develop an online algorithm with the same competitive ration to the one-way trading problem.

The rest of the paper is organized as follows. In Section II, we will introduce a general decoupling framework for the online linear program with online budget entries. In Section III, the decoupling strategy will be applied to solve a general one-way trading problem with online arrival of the budget. Finally, we conclude the paper and highlight future work in Section IV.

II. The Proposed Decoupling Strategy

In this section, we will introduce a general strategy to decouple the online linear program (2) with uncertain budget constraints into multiple online subproblems with offline budget entries. The key in the proposed decoupling strategy is to reformulate Problem (2) into an equivalent problem as follows

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n}\sum_{i=j}^{n} p(i)x(j,i) \\ \text{subject to} & \sum_{i=j}^{n} a(i,k)x(j,i) \le b(j,k)-b(j-1,k), \\ & j=1,2,\dots,n,\ k=1,2,\dots,m. \\ \text{variables} & x(i,j)\in\mathbb{R}^+,\ j\ge i. \end{array} \tag{3}$$

As compared to Problem (2), in Problem (3), we introduce another optimization variable $x(j,i)$ which can be decoupled in different subproblems. Further, the packing constraint is also re-formulated to facilitate the decoupling of the original problem into several subproblems.

The following result demonstrates that Problems (2) and (3) are equivalent, i.e., having an optimal solution for (3), one can readily construct the optimal solution of (2), also, the optimal values for both problems are equal.

Theorem II.1. *Assume $b(j,k)$ is increasing as j increases. Let $x^*(i,j),\ j\ge i$, be an optimal solution for (3). Then, $x^*(i)=\sum_{j=1}^{i} x^*(j,i),\ i=1,2,\dots,n$, is an optimal solution for Problem (2). In addition, the optimal value of Problem (3) is equal to that of Problem (2).*

Proof. First, by rearranging the items, we can rewrite the objective of Problem (3) as $\sum_{i=1}^{n} p(i)\sum_{j=1}^{i} x(j,i)$.

Second, we show that by having a feasible solution to Problem (3), we can construct a feasible solution to Problem (2). Assume that $x(i,j)\in\mathbb{R}^+,\ j\ge i$ is a feasible solution. Then, for $l=1,2,\dots,n$, we have

$$\begin{aligned} &\sum_{i=1}^{l} a(i,k)\sum_{j=1}^{i} x(j,i) \\ \le&\sum_{i=1}^{l}\sum_{j=i}^{l} a(i,k)x(j,i) \\ \le& b(l,k). \end{aligned}$$

Let $x(i)=\sum_{j=1}^{i} x(j,i)$. We have $\sum_{i=1}^{l} a(i,k)x(i)\le b(l,k)$. That is, for each feasible solution $x(i,j)\in\mathbb{R}^+,\ j\ge i$, we can find a feasible solution $x(i)=\sum_{j=1}^{i} x(j,i)$ for problem (2), and the value under $x(i)$ is equal to that of Problem (3) under $x(i,j)$.

Third, we show that for a feasible solution $x(i)$ to Problem (2), we can construct a feasible solution to Problem (3), i.e., $x(i,j)\in\mathbb{R}^+,\ j\ge i$, as follows. Let $x(1,i)=x(i)$ for $i=1,2,\dots,n$, and $x(i,j)=0$ for $i>1$. Obviously, $\sum_{j=1}^{i} x(j,i)=x(i)$ for $i=1,2,\dots,n$. At each step, when we increase/decrease $x(i,j)$ by δ, we must correspondingly decrease/increase $x(l,j)$ $(l\ne i)$ by δ. In this way, we can guarantee that $x(i)=\sum_{j=1}^{i} x(j,i)$ all the time. Then, if there is some j such that $\sum_{i=j}^{n} a(i,k)x(j,i)>b(j,k)-b(j-1,k)$, we have that there must be some l such that $\sum_{i=l}^{n} a(i,k)x(l,i)<b(l,k)-b(l-1,k)$.[1] Otherwise, we have

$$\begin{aligned} b(n,k)=&\sum_{l=1}^{n} b(l,k)-b(l-1,k) \\ <&\sum_{l=1}^{n}\sum_{i=l}^{n} a(i,k)x(l,i) \\ \le&\sum_{i=1}^{n} a(i,k)\sum_{l=1}^{i} x(l,i) \\ =&\sum_{i=1}^{n} a(i,k)x(i). \end{aligned}$$

contradicting the assumption for $x(i)$. In this way, we can always decrease $x(j,i)$ such that $\sum_{i=j}^{n} a(i,k)x(j,i)\le b(j,k)-b(j-1,k)$ for all j.

That is, we can find $x(j,i)$ which satisfies $x(i)=\sum_{j=1}^{i} x(j,i)$ and meanwhile the constraints in Problem (3). Obviously, the value is also equal.

Combining (1) and (2), we have that the optimal values of Problem (2) and (3) are equal. Specifically, when $x^*(i,j),\ j\ge i$ is an optimal solution for Problem (3), $x(i)=\sum_{j=1}^{i} x^*(j,i),\ i=1,2,\dots,n$ is also an optimal solution for Problem (2).

This completes the proof. $\square$

[1] By convention, $b(0,k)$ for any k.

Problem (3) reformulates the original problem and it can be decoupled by solving multiple subproblems. The j-th subproblem is as follows.

$$\begin{array}{ll} \text{maximize} & \sum_{i=j}^{n} p(i)x(j,i) \\ \text{subject to} & \sum_{i=j}^{n} a(i,k)x(j,i) \leq b(j) - b(j-1), \\ & k = 1, 2, \ldots, m. \\ \text{variables} & x(i,j) \in \mathbb{R}^+, \ j \geq i. \end{array} \tag{4}$$

Note that in Problem (4), the value of j is fixed, and its time horizon is from j to n. With fixed j, there is a fixed packing constraint specified by $b(j) - b(j-1)$. In its online version, the entries $p(i)$ and $a(i,k)$ are revealed piece by piece, and the budget entries are available in offline manner at the beginning of the first slot. At time slot i ($i \geq j$), the available information involves $p(l)$ and $a(l,k)$ for $l = j, j+1, \ldots, i$. Based on the above explanation, *each sub-problem can be seen as the classic online packing problem as formulated in* (1).

In the literature, there are extensive research works on the problem where the budget constraint is fixed and known [25], [26].

Now, we explain how to leverage the online algorithms for subproblems and construct an online algorithm for the general case. For the j-th subproblem, assume that there is an online algorithm whose competitive ratio is r_j. We construct the online algorithm for Problem (2) as follows. Let $\hat{x}(i,j), \ (j \geq i)$ be the decision of the j-th algorithm at time slot i. Then, the decision of the constructed general online algorithm at slot i is $\hat{x}(i) = \sum_{j=1}^{i} \hat{x}(j,i)$. Finally, we have the following theorem on the competitiveness of the constructed online algorithms for the general case.

Theorem II.2. *For the online algorithm whose decision at time slot i takes $\hat{x}(i) = \sum_{j=1}^{i} \hat{x}(j,i)$, it achieves the competitive ratio of* $\max_j r_j$.

Proof. First, the profit earned by the online algorithm is

$$\sum_{i=1}^{n} p(i) \sum_{j=1}^{i} \hat{x}(j,i) = \sum_{i=1}^{n} \sum_{j=i}^{n} p(i)\hat{x}(i,j).$$

Let $\mathtt{OPT}$ be the offline optimal value of Problem 3 and $\mathtt{OPT}_i$ be the offline optimal value for Problem 4. We have

$$\begin{aligned} \mathsf{cr} =& \frac{\mathtt{OPT}}{\sum_{i=1}^{n} \sum_{j=i}^{n} p(i)\hat{x}(i,j)} \\ =& \frac{\sum_{i=1}^{n} \mathtt{OPT}_i}{\sum_{i=1}^{n} \sum_{j=i}^{n} p(i)\hat{x}(i,j)} \\ \leq& \max_i r_i. \end{aligned}$$

The second equality is by Theorem II.1. This completes the proof. ☐

The result in above theorem shows that by integrating the online decisions for each subproblem, the competitive ratio of the online linear programming problem with unknown budget entries can attain that for the simple case with fixed constraint. For the case that there is an online algorithm with *optimal*[2] competitive ratio for each subproblem.

Corollary II.3. *If $\hat{x}(i,j), \ (j \geq i)$ is an optimal online solution for the j-th subproblem, the online algorithm whose decision takes $\hat{x}(i) = \sum_{j=1}^{i} \hat{x}(j,i)$ achieves the optimal competitive ratio.*

In many paradigms such as the one-way trading problem, the optimal competitive ratio for the online linear program with uncertain resource constraints can be attained [25]. In the next section, we apply this strategy to construct an optimal online algorithm for the general case of the one-way trading problem.

III. Application to the One-way Trading Problem with Uncertain Budget

The one-way trading problem is a classic paradigm of online linear programming. In the one-way trading problem, one is required to maximize a liner objective $\sum_{i=1}^{n} p(i)x(i)$ while respecting a given constraint $\sum_{i=1}^{n} x(i) \leq b$. The objective is correspondingly the cumulative production of the prices and amount of sold items at each time slot. The entries $p(i), \ i = 1, 2, \ldots, n$, correspond to the market prices over time and $x(i), \ i = 1, 2, \ldots, n$, are the amounts of sold items. Both $p(i)$ and $x(i)$ are nonnegative. The variable $x(i)$ can be either an integer or a fractional number according to different scenarios and the summation of $x(i)$ should be less than or equal to b, as the budget constraint. The offline formulation is

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} p(i)x(i) \\ \text{subject to} & \sum_{i=1}^{n} x(i) \leq b. \\ \text{variables} & x(i) \in \mathbb{R}^+. \end{array} \tag{5}$$

In this paper, and for the online version, we assume the ending time and the constraint (the number of available items) is given and known in advance while the price entries $p(i)$ are unknown and arrive sequentially. Sequential market prices are reasonable in real market, where many complex supply demand relationships work together and influence the dynamics in price. In the online case, the decisions of the designed algorithm are based on only the current and past information on the market price. The optimal algorithm for the one-way trading problem attains a competitive ratio of $O(\ln(U/L))$, where U and L are the upper and lower bound of the price fluctuation, respectively [25]. One can find that the one-way trading problem can be seen as a special case of Problem (2).

Consider the following general one-way trading problem where we assume the revelation of the market prices is online.

[2] An online algorithm with optimal competitive ratio achieves the best possible competitive ratio, i.e., no other algorithm can achieve a better competitive ratio.

In addition to traditional settings, we assume the budget constraints are unknown and time-varying. The offline version of the extended version one-way trading problem is as follows.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} p(i)x(i) \\ \text{subject to} \quad & \sum_{i=1}^{j} x(i) \leq b(j). \\ \text{variables} \quad & x(i) \in \mathbb{R}^{+}. \end{aligned} \tag{6}$$

Note that when $b(j) = b$, the above problem will be reduced to the offline version of the one-way trading problem where the total amount of sold items is fixed.

Now, we apply the proposed decoupling strategy in Section II to tackle Problem (6).

The original allocation problem can be decoupled to the subproblems each of which corresponds to allocating the arriving items at one time slot. Specifically, the arriving budget $b(j) - b(j-1)$ at j-th time slot is allocated among the time slots from j to n by an independent online algorithm as shown in the Appendix A. We use $d(j,i)$, $(i \geq j)$ to denote the sold amount at the j-th time slot for arriving budget at time slot i. The sold amount can be determined by the online algorithm introduced in the Appendix A. In this way, we can guarantee a $\ln(U/L) + 1$ competitive ratio for the subproblem of selling the $b(j) - b(j-1)$ amount of items, where U and L are the maximum and minimum market price, respectively.

By the introduced decoupling strategy, the sold amount at time slot j is set to $\sum_{i=1}^{j} d(i,j)$. The profit earned by the online algorithm is

$$\sum_{j=1}^{n} p(j) \sum_{i=1}^{j} d(i,j) = \sum_{i=1}^{n} \sum_{j=i}^{n} p(j) d(i,j).$$

On the other hand, the optimal profit earned by the offline algorithm is $p_{\max,i}[b(i) - b(i-1)]$, where $p_{\max,i}$ is defined as the maximum price during $[i,n]$. Then, the competitive ratio of the online algorithm satisfies

$$\begin{aligned} \mathsf{cr} \leq & \frac{\sum_{i=1}^{n} p_{\max,i}[b(i) - b(i-1)]}{\sum_{i=1}^{n} \sum_{j=i}^{n} p(j) d(i,j)} \\ \leq & \max_{i=1,2,\ldots,n} \frac{p_{\max,i}[b(i) - b(i-1)]}{\sum_{i=1}^{n} \sum_{j=i}^{n} p(j) d(i,j)} \\ = & \max_{i=1,2,\ldots,n} \mathsf{cr}_i, \end{aligned}$$

where cr_i is the competitive ratio for the i-th subproblem. From the competitive analysis in the appendix, we have that one can guarantee a competitive ratio of $\ln(U/L)+1$ for each subproblem. Thus, the competitive ratio is $\ln(U/L) + 1$ for the general problem.

IV. Conclusion

In this paper, we proposed a novel decoupling strategy to deal with the online linear programming problem with unknown budget entries. By using the proposed decoupling strategy, the general online linear program can be decoupled to several subproblems each with fixed budget constraints given in offline. As for a case study, we applied the proposed strategy to the one-way trading problem with dynamic budget, and achieved the optimal competitive ratio similar to the basic one-way trading problem without dynamic budget. While we presented our results for the maximization problem, the proposed strategy could also be used to tackle minimization problems with online arrival of demand entries.

As for the future work, we plan to tackle even more general online linear programs with inventory constraints, in which the budget constraints are coupled over time.

References

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

[2] Y. Azar, N. Buchbinder, T. H. Chan, S. Chen, I. R. Cohen, A. Gupta, Z. Huang, N. Kang, V. Nagarajan, J. Naor, *et al.*, "Online algorithms for covering and packing problems with convex objectives," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pp. 148–157, IEEE, 2016.

[3] N. Buchbinder, J. S. Naor, *et al.*, "The design of competitive online algorithms via a primal–dual approach," *Foundations and Trends® in Theoretical Computer Science*, vol. 3, no. 2–3, pp. 93–263, 2009.

[4] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[5] L. Yang, L. Deng, M. H. Hajiesmaili, C. Tan, and W. S. Wong, "An optimal algorithm for online non-convex learning," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, p. 25, 2018.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proceedings of the fourth international conference on Future energy systems*, pp. 137–148, 2013.

[8] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2013.

[9] S. Albers and J. Quedenfeld, "Optimal algorithms for right-sizing data centers," in *Proc. of ACM SPAA*, pp. 363–372, 2018.

[10] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-aware online economic dispatching for microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 323–335, 2018.

[11] R. Zhou, Z. Li, and C. Wu, "An online procurement auction for power demand response in storage-assisted smart grids," in *Proc. of IEEE INFOCOM*, pp. 2641–2649, 2015.

[12] M. H. Hajiesmaili, M. Chen, E. Mallada, and C.-K. Chau, "Crowd-sourced storage-assisted demand response in microgrids," in *Proceedings of the Eighth International Conference on Future Energy Systems*, pp. 91–100, ACM, 2017.

[13] B. Alinia, M. H. Hajiesmaili, and N. Crespi, "Online EV charging scheduling with on-arrival commitment," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[14] B. Alinia, M. S. Talebi, M. H. Hajiesmaili, A. Yekkehkhany, and N. Crespi, "Competitive online scheduling algorithms with applications in deadline-constrained EV charging," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, IEEE, 2018.

[15] Z. Mao, C. E. Koksal, and N. B. Shroff, "Optimal online scheduling with arbitrary hard deadlines in multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 177–189, 2016.

[16] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, p. 23, 2017.

[17] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, "Buffer overflow management in qos switches," *SIAM Journal on Computing*, vol. 33, no. 3, pp. 563–583, 2004.

[18] M. H. Hajiesmaili, L. Deng, M. Chen, and Z. Li, "Incentivizing device-to-device load balancing for cellular networks: An online auction design," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 265–279, 2017.

[19] G. Even, M. Medina, and D. Rawitz, "Online generalized caching with varying weights and costs," in *Proc. of ACM SPAA*, pp. 205–212, 2018.

[20] L. Yang, W. S. Wong, and M. H. Hajiesmaili, "An optimal randomized online algorithm for QoS buffer management," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, p. 36, 2017.

[21] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and generalized on-line matching," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 264–273, IEEE, 2005.

[22] A. Mehta *et al.*, "Online matching and ad allocation," *Foundations and Trends® in Theoretical Computer Science*, vol. 8, no. 4, pp. 265–368, 2013.

[23] L. Yang, M. H. Hajiesmaili, R. Sitaraman, E. Mallada, W. S. Wong, and A. Wierman, "Online inventory management with application to energy procurement in data centers," *arXiv preprint arXiv:1901.04372*, 2019.

[24] L. Yang, M. H. Hajiesmaili, H. Yi, and M. Chen, "Hour-ahead offering strategies in electricity market for power producers with storage and intermittent supply," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 45, pp. 21–22, ACM, 2017.

[25] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal search and one-way trading online algorithms," *Algorithmica*, 2001.

[26] N. Buchbinder and J. Naor, "Online primal-dual algorithms for covering and packing," *Mathematics of Operations Research*, vol. 34, no. 2, pp. 270–286, 2009.

[27] S. Agrawal, Z. Wang, and Y. Ye, "A dynamic near-optimal algorithm for online linear programming," *Operations Research*, vol. 62, no. 4, pp. 876–890, 2014.

[28] J. Lorenz, K. Panagiotou, and A. Steger, "Optimal algorithms for k-search with application in option pricing," *Algorithmica*, 2009.

[29] E. Mohr, I. Ahmad, and G. Schmidt, "Online algorithms for conversion problems: a survey," *Surveys in Operations Research and Management Science*, vol. 19, no. 2, pp. 87–104, 2014.

Appendix

A. Optimal Online Algorithms for One-way Trading

In one-way trading problem, the market prices are revealed pieces by piece, so the decisions of the online algorithm can be only based on past information. First, we define the following function

$$g(x) = \begin{cases} L, & x \leq \frac{b}{\ln(U/L)+1}, \\ L \exp\left((\ln(U/L)+1)\frac{x}{b} - 1\right), & \text{otherwise.} \end{cases}$$

The online algorithm maintains a variable P, which records the maximum price observed so far. At each time slot i, if $p(i)$ is larger than P, we set the value if $x(i)$ as follows

$$x(i) = g^{-1}(p(i)) - g^{-1}(P).$$

The profit earned by the online algorithm is correspondingly $p(i)[g^{-1}(p(i)) - g^{-1}(P)]$, which corresponds to the rectangles shown in Figure 1. Let p_i be the price which is larger than P, the profit earned by the online algorithm is just equal to the orange area.

Let $p_{\max}$ be the maximum price during the n time slots, and let x_{end} be the total amount of sold items, we have the profit earned by the online algorithm is at least

$$\int_0^{x_{\text{end}}} g(x)dx.$$

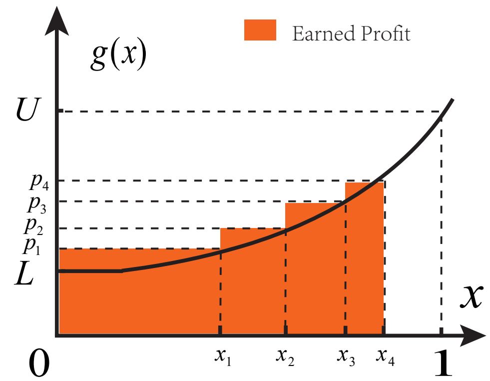

Fig. 1. An illustration of function g and the earned profit by the online algorithm.

On the other hand, the profit earned by the optimal offline algorithm is at most $bp_{\max}$. According to the features of g function, we have

$$\begin{aligned} &\frac{p_{\max} b}{\int_0^{x_{\text{end}}} g(x)dx} \\ =&\frac{Lb\exp\left(\frac{(\ln(U/L)+1)x_{\text{end}}}{b} - 1\right)}{\frac{Lb}{\ln(U/L)+1} + \frac{Lb}{\ln(U/L)+1}\exp\left(\frac{(\ln(U/L)+1)x_{\text{end}}}{b} - 1\right) - \frac{Lb}{\ln(U/L)+1}} \\ =&\ln(U/L) + 1. \end{aligned}$$

That means that the above online algorithm is $(\ln(U/L)+1)$-competitive, achieving the optimal performance.