SIGACT News Online Algorithms Column 25

Rob van Stee
University of Leicester
Leicester, United Kingdom

For this last column of 2014, Zhiyi Huang agreed to write about the primal-dual framework for online algorithms. This topic was previously discussed in this column back in 2007, which gave a general introduction to this type of design and analysis. The current contribution brings us up to date with recent developments in this area. Thank you very much for your contribution, Zhiyi!

As always, I would like to invite more contributions to this column, be it surveys, conference reports, or technical articles related to online algorithms and competitive analysis. If you are considering becoming a guest writer, don't hesitate to mail me at rob.vanstee@le.ac.uk.

SIGACT News Online Algorithms Column 25:
Online Primal Dual: Beyond Linear Programs

Zhiyi Huang*

1 Introduction

Due to the wide range of applications of online combinatorial optimization, developing techniques for designing and analyzing online algorithms has been an active area in theoretical computer science for decades. In this paper, we will focus on the online primal dual technique developed by Buchbinder and Naor [10]. Informally, the online primal dual technique designs and analyzes online algorithms based on linear program relaxations and duality. Linear programs and duality are one of the most fundamental tool in (offline) combinatorial optimization and online primal dual can be viewed as their online counterpart. This technique has given nearly optimal online algorithms for many problems, including online paging and caching [3, 5], online packing and covering [11], online matching [12, 16], etc.

In this paper, we will talk about some recent research on extending the online primal dual technique beyond linear programs. The extension allows us to apply the technique to problems that do not have natural linear program relaxations. For example, consider the problem of scheduling jobs on speed-scalable machines that can run at different speeds subject to different energy costs. Convex programs can naturally model energy costs while not finite-size linear programs can do so.

Furthermore, it allows us to design online algorithms with better competitive ratios for problems with linear program relaxations by relaxing some of the linear constraints with convex cost functions in the objective. The hardness of some online problems, e.g., online combinatorial auction, comes from having a fixed amount of resources while not knowing the number of online requests that demand the resources. In many practical scenarios, however, it is possible to get more resources subject to extra costs. Mathematically, this is equivalent to replacing the linear resource constraints in the linear program relaxation with convex cost functions in the objective. The problem with relaxed resource constraints often admits a better competitive ratio (with nice-behaving cost functions) than the original problem. For example, the competitive ratio of online combinatorial auction with hard resource constraints is at least logarithmic [6, 8], while the relaxed problem with, e.g., polynomial cost functions, admits constant competitive online algorithms [7, 18].

In the rest of this paper, we will focused on convex programs and Fenchel duality in the non-stochastic setting. Readers are also referred to other interesting work along this line, including using the online primal dual technique (or dual fitting) with Lagrangian duality [2, 17, 22], and online stochastic convex optimization [1].

Organization We will first recap the online primal dual technique (Section 2), and explain its extension to convex programs (Section 3). Then, we will talk about three applications of the

*Department of Computer Science, the University of Hong Kong. Supported in part by the Research Grant Council of Hong Kong (HKU272002/14E). Email: zhiyi@cs.hku.hk.

extension: online combinatorial auction with production costs [18] (Section 4), online matching with concave returns [15] (Section 5), and online scheduling with speed scaling [14] (Section 6).

2 Online Primal Dual in a Glimpse

We will briefly recap the online primal dual technique for linear programs by demonstrating its application on the ski rental problem (e.g., [9, 12]). In the ski rental problem, a skier arrives at a ski resort but does not know the number of days of his stay. Every day, the skier needs to decide whether to rent skis at a cost of \$1 or to buy skis at a cost of \$$B$. We assume for simplicity that B is a positive integer. The goal is to minimize the total cost.

We present the primal and dual linear programs of the ski rental problem in Figure 1. x is the indicator of whether the algorithm buys skis, and y_t is the indicator of whether the algorithm rent skis on day t. For simplicity, we will discuss solving the linear programs online to minimize the expected primal objective value. Readers are referred to the survey by Buchbinder and Naor [9, Section 3] for an online rounding algorithm which convert the online fractional solution into a randomized integral algorithm of the ski rental problem with the same competitive ratio. Roughly speaking, the fractional value of x denotes the probability that the algorithm would buy skis and the fractional value of y_t denotes the probability that the algorithm would rent skis on day t.

$$\begin{array}{lll} \underset{x,y \geq 0}{\text{maximize}} & B \cdot x + \sum_{t=1}^{T} y_t & \\ \text{subject to} & x + y_t \geq 1 & t = 1, \dots, T \end{array} \qquad \begin{array}{lll} \underset{\alpha \geq 0}{\text{minimize}} & \sum_{t=1}^{T} \alpha_t & \\ \text{subject to} & \alpha_t \leq 1 & t = 1, \dots, T \\ & \sum_{t=1}^{T} \alpha_t \leq B & \end{array}$$

Figure 1: Primal and dual linear programs of the ski rental problem

Online Primal Dual Framework As the input data arrives over time, more variables and constraints of the primal and dual programs are revealed. On each day t, a new primal variable $y_t \geq 0$, a new primal constraint $x + y_t \geq 1$, and a new dual variable $1 \geq \alpha_t \geq 0$ arrive. Online primal dual algorithms maintain a feasible primal assignment and a feasible dual assignment simultaneously. Let $\Delta_t P$ and $\Delta_t D$ be the changes of the primal and dual objectives respectively on day t. The goal is to satisfy

$$\Delta_t P \leq F \cdot \Delta_t D \tag{1}$$

for some fixed parameter $F \geq 1$. If (1) holds, then the algorithm is F competitive because the final primal objective P is at most F times that of the final dual objective D, and by weak duality D is less than or equal to the optimal primal objective.

Relaxed Complementary Slackness It is known that in the offline primal and dual solutions, the primal/dual variables and the corresponding dual/primal constraints satisfy complementary slackness (e.g., [13]), i.e., a primal/dual variable must be zero unless the corresponding dual/primal constraint is tight. Specifically,

(a) x must be zero unless $\sum_{t=1}^{T} \alpha_t = B$;

(b) y_t must be zero unless $\alpha_t = 1$;

(c) α_t must be zero unless $x + y_t = 1$.

In the online problem, however, it is not possible to satisfy all complementary slackness conditions exactly throughout (in particular, condition (a) and (c)). Online primal dual algorithms are therefore driven by satisfying these conditions approximately, where the value of a primal/dual variable depends on the tightness of the corresponding dual/primal constraint. Concretely, consider the following relaxed conditions (condition (b) remains the same):

(a') x depends on the tightness of $\sum_{t=1}^{T} \alpha_t \leq B$, i.e., is a monotone function of $\sum_{t=1}^{T} \alpha_t$;

(c') α_t must be zero unless $x + y_t = 1$ at time t (the constraint may have slack in the future because the algorithm may increase x).

Online Primal Dual Algorithms from Relaxed Complementary Slackness On the $\hat{t}$-th day, $\hat{t} = 1, \ldots, T$, first consider the new dual variable $\alpha_{\hat{t}}$. To maximize the dual objective, let $\alpha_{\hat{t}} = \min\{1, B - \sum_{t=1}^{\hat{t}-1} \alpha_t\}$. Recall that B is an integer, this is equivalent to letting $\alpha_{\hat{t}} = 1$ if $\hat{t} \leq B$ and $\alpha_{\hat{t}} = 0$ otherwise. As a result, $\sum_t \alpha_t$ increases by 1 on each day $\hat{t} \leq B$ and the algorithm increases x accordingly (condition (a')). Let $x_{\hat{t}}$ be the value of x after day $\hat{t}$ for $\hat{t} = 1, \ldots, B$, and $x_0 = 0$. After day B, $\sum_t \alpha_t$ and, thus, the value of x remain constant. Further, let $y_{\hat{t}} = 1 - x_{\hat{t}}$ (condition (c') and to minimize the primal objective). Finally, since we let $\alpha_{\hat{t}} = 0$ for $\hat{t} > B$, we must have $y_{\hat{t}} = 0$ and $x = 1$ on any day $\hat{t} > B$ (condition (b)) and, thus, $x_B = 1$.

In sum, for every monotone sequence x_t, $t = 0, 1, \ldots, B$, such that $x_0 = 0$ and $x_B = 1$, there is an online primal dual algorithm as follows:

1. On day $\hat{t} = 1, \ldots, B$, let $\alpha_{\hat{t}} = 1$, $x = x_{\hat{t}}$, and $y_{\hat{t}} = 1 - x_{\hat{t}}$.
2. On day $\hat{t} > B$, let $\alpha_{\hat{t}} = 0$, x remains the same (i.e., equals 1), and $y_{\hat{t}} = 0$.

Online Primal Dual Analysis It remains to find the best sequence $\{x_t\}_{t=1,\ldots,B}$ such that (1) holds with the smallest possible $F \geq 1$. Note that both the primal and dual objectives remain the say after day B, so it suffices to analyze (1) for the first B days. On each day $t = 1, \ldots, B$, x changes from x_{t-1} to x_t, and $y_t = 1 - x_t$. So the change of primal objective is

$$\Delta_t P = B \cdot (x_t - x_{t-1}) + (1 - x_t) = (B - 1) \cdot x_t - B \cdot x_{t-1} + 1 \ .$$

On the other hand, the algorithm sets $\alpha_t = 1$. So the change of dual objective is

$$\Delta_t D = 1 \ .$$

So (1) becomes $(B - 1) \cdot x_t - B \cdot x_{t-1} \leq F - 1$. Reorganizing terms, we have that $x_t + F - 1 \leq \frac{B}{B-1}(x_{t-1} + F - 1)$ and, thus,

$$x_t \leq \left(\tfrac{B}{B-1}\right)^t (x_0 + F - 1) - (F - 1) = \left(\left(\tfrac{B}{B-1}\right)^t - 1\right)(F - 1) \tag{2}$$

Let $e(B) = \left(\frac{B}{B-1}\right)^B$. We have $e(B) \geq e$ and $\lim_{B \to +\infty} e(B) = e \approx 2.718$. By $x_B = 1$ and the above inequality, we have $F \geq \frac{e(B)}{e(B)-1}$. Let $F = \frac{e(B)}{e(B)-1}$ and choose x_t such that (2) holds with equality. Then, we have an online primal dual algorithm with competitive ratio $\frac{e(B)}{e(B)-1} \leq \frac{e}{e-1} \approx 1.582$.

3 Convex Programs and Fenchel Duality

Conjugate Let $f : \mathbb{R}^n_+ \mapsto \mathbb{R}$ be a convex function. Its *convex conjugate* is defined as

$$f^*(x^*) = \operatorname*{maximize}_{x \geq 0} \left\{ \langle x, x^* \rangle - f(x) \right\} \ .$$

For example, if $f(x) = \frac{1}{\alpha} x^\alpha$ is a polynomial, then $f^*(x^*) = (1 - \frac{1}{\alpha}) {x^*}^{\frac{\alpha}{\alpha - 1}}$ is also a polynomial. For simplicity, we will further assume that f is non-negative, non-decreasing, strictly convex, and differentiable for the rest of this paper. In this case, the conjugate satisfies the following properties:

- f^* is non-negative, non-decreasing, strictly convex, and differentiable;
- $f^{**} = f$;
- ∇f and ∇f^* are inverse of each other, and we say that x and x^* form a complementary pair if $x = \nabla f^*(x^*)$ and $x^* = \nabla f(x)$;

Next, consider a concave function $g : \mathbb{R}^n_+ \mapsto \mathbb{R}$. Its *concave conjugate* is defined as

$$g_*(x^*) = \operatorname*{minimize}_{x \geq 0} \left\{ \langle x, x^* \rangle - g(x) \right\} \ .$$

Similar to the convex case, we will further assume g to be non-negative, non-decreasing, strictly convex, and differentiable for the rest of this paper. In this case, the concave conjugate satisfies the following properties:

- g_* is non-positive, non-decreasing, strictly convex, and differentiable;
- $g_{**} = g$;
- ∇g and ∇g^* are inverse of each other, and we say that x and x^* form a complementary pair if $x = \nabla g_*(x^*)$ and $x^* = \nabla g(x)$;

Assuming mild regularity conditions which we omit in this paper, the following strong duality holds:

$$\begin{aligned} \operatorname*{minimize}_{x \geq 0} \left\{ f(x) - g(x) \right\} &= \operatorname*{maximize}_{x^* \geq 0} \operatorname*{minimize}_{x \geq 0} \left\{ f(x) - \langle x, x^* \rangle + \langle x, x^* \rangle - g(x) \right\} \\ &= \operatorname*{maximize}_{x^* \geq 0} \left\{ g_*(x^*) - f^*(x^*) \right\} \ . \end{aligned}$$

Fenchel Duality of Convex Programs with Linear Constraints In this paper, we will restrict our attention to convex programs with linear constraints. In the rest of this section, we will consider online combinatorial auction with production costs as an illustrative example, and will explain its convex program relaxation and the dual convex program.

Let there be a seller with m items for sale, and n buyers that arrive online. Each buyer i has a combinatorial valuation function $v_i : 2^{[m]} \mapsto \mathbb{R}_+$. On the arrival of a buyer i, the seller allocates a bundle of items $S \subseteq [m]$ to the buyer. The seller may produce an arbitrary number of copies of each item j subject to a production cost function f_j (i.e., producing y_j copies costs $f_j(y_j)$). The goal is to maximize the social welfare, i.e., the sum of each buyer's value for the allocated bundle of items less the total production cost.

Below is the convex program relaxation of online combinatorial auction with production costs (assume for simplicity that f_j's are defined for all non-negative real numbers):

$$
\begin{array}{lll}
\underset{x,y\geq 0}{\text{maximize}} & \sum_{i=1}^{n}\sum_{S\subseteq[m]} v_i(S)x_{iS} - \sum_{j=1}^{m} f_j(y_j) & \\
\text{subject to} & \sum_{S\subseteq[m]} x_{iS} \leq 1 & i \in [n] \\
& \sum_{i=1}^{n}\sum_{S\ni j} x_{iS} \leq y_j & j \in [m]
\end{array}
\tag{3}
$$

Here, x_{iS} is the indicator of whether buyer i gets bundler S. y_j is the total demands of item j.

The Fenchel dual convex program can be derived from the Lagrangian dual program and the definition of convex conjugate. Taking the Lagrangian dual, we have

$$
\begin{aligned}
\underset{u,p\geq 0}{\text{minimize}}\ \underset{x,y\geq 0}{\text{maximize}} \quad & \sum_{i=1}^{n}\sum_{S\subseteq[m]} v_i(S)x_{iS} - \sum_{j=1}^{m} f_j(y_j) \\
& + \sum_{i=1}^{n} u_i \cdot \big(1 - \sum_{S\subseteq[m]} x_{iS}\big) + \sum_{j=1}^{m} p_j \cdot \big(y_j - \sum_{i=1}^{n}\sum_{S\ni j} x_{iS}\big)
\end{aligned}
$$

First, consider the maximization problem w.r.t. x_{iS}, namely,

$$
\underset{x_{iS}\geq 0}{\text{maximize}}\ \big(v_i(S) - u_i - \sum_{j\in S} p_j\big) \cdot x_{iS} = \begin{cases} 0 & \text{if } u_i + \sum_{j\in S} p_j \geq v_i(S), \\ +\infty & \text{otherwise.} \end{cases}
$$

Thus, it imposes a linear constraint $u_i + \sum_{j\in S} p_j \geq v_i(S)$ on the dual problem. Next, consider the maximization problem w.r.t. y_j, namely,

$$
\underset{y_j\geq 0}{\text{maximize}}\ \ p_j y_j - f_j(y_j)
$$

By the definition of convex conjugate, the optimal value of the above maximization problem is $f_j^*(p_j)$. In sum, the Lagrangian dual can be simplified as follows:

$$
\begin{array}{lll}
\underset{u,p\geq 0}{\text{minimize}} & \sum_{i=1}^{n} u_i + \sum_{j=1}^{m} f_j^*(p_j) & \\
\text{subject to} & u_i + \sum_{j\in S} p_j \geq v_i(S) & i \in [n], S \subseteq [m]
\end{array}
\tag{4}
$$

The above convex program is the Fenchel dual convex program of combinatorial auction with production costs. We can interpret p_j as the price of item j and u_i as the utility of buyer i (i.e., his value for the allocated bundle less the total price).

Online Primal Dual with Convex Programs Online primal dual algorithms for convex programs are driven by the Karush-Kuhn-Tucker (KKT) conditions [19, 20]. The KKT conditions for linear constraints are the same as complementary slackness. Online primal dual algorithms handle variables only involved in linear constraints in the same way as in the original approach for linear programs. Next, consider variables involved in the convex functions in the primal and dual objectives, namely, y_j and p_j. The KKT conditions indicate that in the offline primal and dual solutions, y_j and p_j form a complementary pair, i.e., $p_j = f_j'(y_j)$, for $j = 1, \ldots, m$. This coincides with the economic intuition, namely, the unit price of an item j shall equal to the marginal production cost of the item. In the online problem, however, the algorithm knows only the current supply y_j of each item j, but not the final supply. As a result, the algorithm needs to predict the final supply according to the current supply (e.g., double the current supply), and set the price according to the predicted final supply (e.g., $p_j = f'(2y_j)$). To minimize the competitive ratio, let $p_j = p_j(y_j)$ be a monotone function of y_j to be determined. We will choose the pricing functions to minimize $F \geq 1$ such that the change of the dual objective is at most F times the change of the primal objective as the algorithm proceeds (similar to (1)).

4 Example 1: Online Combinatorial Auction with Production Costs

In this section, we will describe the online primal dual algorithms and their analysis for the illustrative example in the previous section, namely, online combinatorial auction with production costs. The results are from a paper by Huang and Kim [18].

Recall that the primal and dual convex programs are given in (3) and (4) respectively. For simplicity, we assume all f_j's are the same and denote it and its convex conjugate as f and f^*. The results extend to heterogeneous f_j's where the competitive ratio depends on the "worst" production cost function f_j.

4.1 Online Primal Dual Algorithms

First consider solving the fractional problem, namely, allowing x_{iS}'s to take fractional values between 0 and 1. The online primal dual algorithms are driven by the following relaxed KKT conditions and a regularity condition.

Relaxed KKT Conditions:

(a) x_{iS} must be zero unless $u_i + \sum_{j \in S} p_j = v_i(S)$ when buyer i comes (the constraint may have slack in the future because the algorithm may increase p_j's);

(b) $p_j = p(y_j)$ is a function of the current demand y_j.

Regularity Condition:

(c) $y_j = \sum_{i=1}^{n} \sum_{S \ni j} x_{iS}$, i.e., the algorithm should not produce more than the current demand.

Given any pricing function $p : \mathbb{R}_+ \mapsto \mathbb{R}_+$, we have the online primal dual algorithm in Figure 2.

1. Initially, let $y_j = 0$ and $p_j = 0$ for $j = 1, \ldots, m$.
2. Maintain $p_j = p(y_j)$ (condition (b)) and $y_j = \sum_{i=1}^{n} \sum_{S \ni j} x_{iS}$ (condition (c)).
3. On the arrival of buyer i, continuously increase x_{iS}'s as follows until $\sum_{S \subseteq [m]} x_{iS} = 1$:
   (a) Let $S^* = \arg\min_{S \subseteq [m]} \left\{ v_i(S) - \sum_{j \in S} p_j \right\}$, breaking ties arbitrarily.
   (b) Increase x_{iS^*} by dx and u_i by $\left(v_i(S^*) - \sum_{j \in S^*} p_j \right) \cdot dx$ (condition (a)).
   (c) Update y_j's and, thus, p_j's accordingly.

Figure 2: Online primal dual algorithm for online combinatorial auction with production cost function f and price function p

4.2 Online Primal Dual Analysis

Following the same framework online primal dual analysis as in the case of linear programs, we seek to prove $F \cdot dP \geq dD$ for some fixed parameter $F \geq 1$, where dP and dD denotes the changes of the primal and dual objectives as the algorithm continuously increases x_{iS}'s. In particular, Huang and Kim [18] showed that the best pricing function and the corresponding parameter F is derived from solving a differential equation:

Theorem 4.1. *If the function p satisfies the following differential equation*

$$\forall y \geq 0: \quad F \cdot \big(p(y) - f'(y)\big) \cdot dy \geq f^{*\prime}(p(y)) \cdot dp(y) \quad \textit{subject to} \quad p(0) = 0,\, p(y) \geq 0 \tag{5}$$

for some fixed parameter $F \geq 1$, then the online primal dual algorithm in Figure 2 is F-competitive for fractional combinatorial auction with production cost function f.

For illustrative purposes, we present a proof sketch of the above theorem. We will omit the proofs of similar theorems in the other two examples.

Proof. Let i be the current buyer and suppose the algorithm has increased x_{iS^*} by dx and has updated the other primal and dual variables accordingly. It suffices to show $F \cdot dP \geq dD$.

First, consider the change of the primal objective:

$$dP = v_i(S^*) \cdot dx - \textstyle\sum_{j \in S} f'(y_j) \cdot dy_j \ .$$

Further, by the definition of the algorithm, dy_j equals dx for $j \in S^*$ and equals 0 for $j \notin S^*$, and that $v_i(S^*) \cdot dx = du_i + \sum_{j \in S^*} p_j \cdot dx = du_i + \sum_{j=1}^m p_j \cdot dy_j$. The change of primal objective is

$$dP = du_i + \textstyle\sum_{j=1}^m p_j \cdot dy_j - \sum_{j=1}^m f'(y_j) \cdot dy_j$$

Further, the change of dual objective is:

$$dD = du_i + \textstyle\sum_{j=1}^m f^{*\prime}(p_j) \cdot dp_j$$

Therefore, to show $F \cdot dP \geq dD$, it suffices to show

$$F \cdot \big(du_i + \textstyle\sum_{j=1}^m p_j \cdot dy_j - \sum_{j=1}^m f'(y_j) \cdot dy_j\big) \geq du_i + \sum_{j=1}^m f^{*\prime}(p_j) \cdot dp_j \ ,$$

which follows from $du_i \geq 0$, $F \geq 1$, and (5). □

Huang and Kim [18] further showed that the best possible competitive ratio is indeed characterized by differential equation (5) and the online primal dual algorithm in Figure 2 is optimal.

Theorem 4.2. *If there is an F-competitive online algorithm for fractional combinatorial auction with production cost function f, then there is a feasible solution for differential equation (5).*

4.3 Applications

So far, we have focused on solving the fractional problem online. Huang and Kim [18] showed that the pricing functions and online primal dual algorithms for the fractional problem can also be used to solve the integral problem for many natural production cost functions with minor changes and essentially the same competitive ratios. In particular, for several previously studied families of production cost functions, including polynomial functions [7], functions with concave derivatives [7], and zero-infinity step functions (i.e., fixed supply) [6, 8], the competitive ratios of the aforementioned online primal dual algorithms are nearly optimal, matching or improving those of previous work.

Theorem 4.3. *For any $\epsilon > 0$, there is an $(e\alpha + \epsilon)$-competitive online primal dual algorithm for (integral) combinatorial auction with polynomial production cost functions $f(y) = y^\alpha$, and there are no $(e\alpha - \epsilon)$-competitive online algorithms.*

Theorem 4.4. *For any $\epsilon > 0$, there is a $(4+\epsilon)$-competitive online primal dual algorithm for (integral) combinatorial auction with production cost function f such that f' is concave, and there are no $(4-\epsilon)$-competitive online algorithms.*

Theorem 4.5. *There is an $O\big(\log m + \log \frac{v_{\max}}{v_{\min}}\big)$-competitive online primal dual algorithm for (integral) combinatorial auction with $\Omega\big(\log m + \log \frac{v_{\max}}{v_{\min}}\big)$ copies of each item, and there are no $o\big(\frac{\log m}{\log\log m} + \log \frac{v_{\max}}{v_{\min}}\big)$-competitive online algorithms.*

5 Example 2: Online Matching with Concave Returns

In this section, we will talk about online matching with concave returns introduced by Devanur and Jain [15]. We will modify the original presentation of the results and analysis for consistency with the other parts of this paper. The underlying ideas and reasonings are the same.

Let there be m agents that are known upfront, and n items that arrive online. On the arrival of an item, the online algorithm needs to match the item to one of the agents immediately. Let v_{ij} be agent j's value for item i. Each agent j is associated with a concave, non-decreasing function $g_j : \mathbb{R}_+ \mapsto \mathbb{R}_+$ such that agent j's value for a bundle of item $S \subseteq [n]$ is $g_j(\sum_{i \in S} v_{ij})$. In this sense, g_j serves as a discount function of agent j's value for receiving multiple items. The goal is to maximize the total value of the matching.

For simplicity, we assume that g_j's are the same and denote it and its concave conjugate as g and g_*. We further assume that $g'(0) = 1$, i.e., the value would not discounted if no items have been allocated to the agent so far.

The convex program relaxation and its dual are given below in Figure 3:

$$\begin{array}{lll} \underset{x,y \geq 0}{\text{maximize}} & \sum_{j=1}^m g(y_j) & \\ \text{subject to} & \sum_{i=1}^n x_{ij} \leq 1 & j \in [m] \\ & y_j \leq \sum_{j=1}^m v_{ij} x_{ij} & i \in [n] \end{array} \qquad \begin{array}{lll} \underset{\alpha \geq 0}{\text{minimize}} & \sum_{i=1}^n \alpha_i - \sum_{j=1}^m g_*(\beta_j) & \\ \text{subject to} & \alpha_i \geq v_{ij}\beta_j & i \in [n], j \in [m] \end{array}$$

Figure 3: Primal and dual convex programs of online matching with concave return function g

5.1 Online Primal Dual Algorithms

Again, we will focus on solving the fractional problem, namely, allowing x_{ij}'s to take fractional values between 0 and 1. The online primal dual algorithms are driven by the following relaxed KKT conditions and a regularity condition.

Relaxed KKT Conditions:

(a) x_{ij} must be zero unless $\alpha_i = v_{ij}\beta_j$ when item i comes (the constraint may have slack in the future because the algorithm may decrease β_j's);

(b) $\beta_j = \beta(y_j)$ is a function of the current total value y_j.

Regularity Condition:

(c) $y_j = \sum_{i=1}^{n} v_{ij} x_{ij}$.

Here, β_j serves as a discount factor of the value of allocating an item to agent j. In the offline primal and dual solutions, β_j and y_j forms a complementary pair and, thus, β_j is equal to the derivative of g at agent j's final total value y_j. In the online problem, however, the algorithm can only predict the final total value based on the current total value. Therefore, we let $\beta_j = \beta(y_j)$ where β is a discount function to be determined later, which specifies how much the algorithm discounts an agent's value from future items given his current total value. Recall that g is concave and $g'(0) = 1$. The value of the discount function starts from $\beta(0) = g'(0) = 1$ and decreases as the agent's total value increases. Given any discount function $\beta : \mathbb{R}_+ \mapsto \mathbb{R}_+$, we have the online primal dual algorithm in Figure 4.

1. Initially, let $y_j = 0$ and $\beta_j = g'(0) = 1$ for $j = 1, \ldots, m$.
2. Maintain $\beta_j = \beta(y_j)$ (condition (b)) and $y_j = \sum_{i=1}^{n} v_{ij} x_{ij}$ (condition (c)) throughout.
3. On the arrival of each item i, continuously increase x_{ij}'s as follows until $\sum_{j=1}^{m} x_{ij} = 1$:
   (a) Let $j^* = \arg\max_{j \in [m]} v_{ij}\beta_j$, breaking ties arbitrarily.
   (b) Increase x_{ij^*} by dx and α_i by $v_{ij^*}\beta_{j^*} \cdot dx$ (condition (a)).
   (c) Update y_j's and, thus, β_j's accordingly.

Figure 4: Online primal dual algorithm for online matching with concave return function g and discount function β

5.2 Online Primal Dual Analysis

Recall that the online primal dual analysis seeks to show $F \cdot dP \geq dD$ for some fixed parameter $F \geq 1$, where dP and dD are the changes of the primal and dual objectives as the algorithm continuously increase x_{ij}'s. Similar to the case of online combinatorial auction with production costs, Devanur and Jain [15] showed that the best discount function β and the corresponding parameter F can be derived from solving a differential equation.

Theorem 5.1. *If the discount function β satisfies the following differential equation*

$$\forall y \geq 0 : \quad F \cdot g'(y) \cdot dy \geq \beta(y) \cdot dy - g'_*\big(\beta(y)\big) \cdot d\beta(y) \quad \textit{subject to} \quad \beta(0) = g'(0) = 1,\ \beta(y) \geq 0 \tag{6}$$

for some fixed parameter $F \geq 1$, then the online primal dual algorithm in Figure 4 is F-competitive for fractional online matching with concave return function g.

Devanur and Jain [15] further showed a matching lower bound:

Theorem 5.2. *If there is an F-competitive online algorithm for fractional online matching with concave return function g, then there is a discount function β that satisfies (6).*

5.3 Applications

Considering concave return functions in online bipartite matching problems is motivated by the adword problem. A concrete motivation is the budget constraint of advertisers: an advertiser would pay his bid for each allocated ad slot but there is a hard budget constraint on his maximum payment per day. As a result, the revenue (return function) is a budget additive function $g(y) = \min\{y, B\}$ for some budget $B > 0$. For this special case, we can explicitly find the smallest possible $F = \frac{e}{e-1}$, and solve differential equation (6), i.e., $\beta(y) = \frac{1}{e-1}\left(e - e^{y/B}\right)$.

Theorem 5.3. *There is an $\frac{e}{e-1}$-competitive online algorithm for the adword problem under the small-bid assumption, which is equivalent to fractional online matching with budget additive concave return function $g(y) = \min\{y, B\}$ for some budget $B > 0$.*

The competitive ratio in the above theorem is optimal, matching the ratio from previous work (e.g., Buchbinder et al. [12], Mehta et al. [21]).

6 Example 3: Online Scheduling with Speed Scaling

Let there be m machines that are known upfront, and n jobs that arrive online. Each machine j can run at different speeds subject to different energy costs. Let f_j be the power function of machine j such that running the machine at speed y_j consumes $f_j(y_j)$ energy per unit of time. Each job i is defined by its arrival time r_i, weight w_i, and volumes v_i. A feasible schedule consists of an allocation of jobs to time slots on machines and a speed profile specifying how fast the machines run at each time slot, such that the total computational resources assigned to the job, i.e., the sum of speeds of the allocated time slots, is at least its volume.

There are two natural objectives in this problem: minimizing total energy consumption, and minimizing the delay of jobs. A standard measure of the delay of jobs is flow time. The flow time of a job is the difference between its completion time and arrival time. The weighted flow time is the weighted sum of the flow time of jobs. In this section, we focus on minimizing the sum of weighted flow time and energy.

This problem is more complicated than the previous two examples in the sense that the online algorithm needs to make multiple types of online decisions. At each time slot, the online algorithm needs to decide which job to run on each machine (*job selection*) and at what speed (*speed scaling*). Further, on the arrival of a job, the online algorithm needs to assign the job to one of the machines immediately (*job assignment*). Note that the algorithm may preempt the current job processing on a machine with the new job and resume after the new job is finished. This usually happens when the new job is more important, e.g., having larger weight/volume ratio.

The main results in this section are from a paper by Devanur and Huang [14]. For simplicity, we will present the algorithms and their analysis for the case of a single machine while the results hold more generally for multiple unrelated machines (i.e., a job can have distinct weights and volumes on different machines).

The convex program relaxation of the single machine problem and its dual is given in Figure 5. Here, x_{it} is the speed of the machine at time t on processing job i. y_t is the total speed at time t.

We will first discuss a simple online primal dual algorithm and its analysis to illustrate the structure of the convex programs. Then, we will explain how to improve the simple algorithm to achieve asymptotically optimal competitive ratios.

$$\begin{aligned}
\underset{x,y\geq 0}{\text{minimize}} \quad & \textstyle\sum_{i=1}^{n} \int_{r_i}^{+\infty} w_i(t-r_i)x_{it}dt + \int_0^{+\infty} f(y_t)dt & \\
\text{subject to} \quad & \textstyle\int_{r_i}^{+\infty} x_{it} \geq v_i & i \in [n] \\
& y_t \geq \textstyle\sum_{i=1}^{n} x_{it} & t \geq 0
\end{aligned}$$

$$\begin{aligned}
\underset{\alpha,\beta\geq 0}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{n} v_i\alpha_i - \int_0^{+\infty} f^*(\beta_t)dt & \\
\text{subject to} \quad & \alpha_i \leq \beta_t + \frac{w_i}{v_i}(t-r_i) & i \in [n], t \geq r_i
\end{aligned}$$

Figure 5: Primal and dual convex programs of online scheduling for minimizing fractional flow time plus energy with power function f on a single machine

6.1 A Simple Online Primal Dual Algorithm

Again, we will focus on the fractional problem, namely, solving the primal convex program in Figure 5 online. The fractional problem is known as minimizing fractional flow time plus energy. We first present the relaxed KKT conditions and a regularity condition that lead to the online primal dual algorithms.

Relaxed KKT Conditions:

(a) x_{it} must be zero unless $\alpha_i = \beta_t + \frac{w_i}{v_i}(t - r_i)$ when job i comes (the constraint may have slack in the future because the algorithm may increase β_t's);

(b) $\beta_t = \beta(y_t)$ is a function of the current total speed y_t.

Regularity Condition:

(c) $y_t = \sum_{i=1}^{n} x_{it}$.

Given any function $\beta : \mathbb{R}_+ \mapsto \mathbb{R}_+$, we have the online primal dual algorithm in Figure 6.

1. Initially, let $y_t = 0$ and $\beta_t = f'(0)$ for $j = 1, \ldots, m$, $t \geq 0$.
2. Maintain $\beta_t = \beta(y_t)$ (condition (b)) and $y_t = \sum_{i=1}^{n} x_{it}$ (condition (c)) throughout.
3. On the arrival of each job i, continuously increase x_{it}'s as follows until $\int_{r_i}^{+\infty} x_{it} = v_i$:
   (a) Let $t^* = \arg\max_{t\geq r_i} \left\{\beta_t + \frac{w_i}{v_i}(t - r_i)\right\}$, breaking ties arbitrarily.
   (b) Increase x_{it^*} by dx and α_i by $\left(\beta_{t^*} + \frac{w_i}{v_i}(t^* - r_i)\right) \cdot dx$ (condition (a)).
   (c) Update y_t's and, thus, β_t's accordingly.

Figure 6: A simple online primal dual algorithm for online scheduling for minimizing fractional flow time plus energy with power function f on a single machine

6.2 Online Primal Dual Analysis of the Simple Algorithm

Recall that the online primal dual analysis seeks to show $F \cdot dP \geq dD$ for some fixed parameter $F \geq 1$, where dP and dD are the changes of the primal and dual objectives as the algorithm continuously increase x_{it}'s.

Theorem 6.1. *If β satisfies the following differential equation*

$$\forall y \geq 0 : \quad \big(F \cdot \beta(y) - f'(y)\big) \cdot dy \geq F \cdot f^{*\prime}(\beta(y)) \cdot d\beta(y) \quad \text{subject to} \quad \beta(0) = g'(0),\, \beta(y) \geq 0 \quad (7)$$

for some fixed parameter $F \geq 1$, then the online primal dual algorithm in Figure 6 is F-competitive for minimizing fractional flow time plus energy with power function f on a single machine.

The most common power functions in the literature are polynomial power functions with degree 2 or 3. For the special case of polynomial power functions, we can explicitly solve differential equation (7) and find the smallest F for which the differential equation is feasible.

Theorem 6.2. *For polynomial power function $f(y) = y^\alpha$, then the online primal dual algorithm in Figure 6 with $\beta(y) = \frac{y^{\alpha-1}}{\alpha^{\alpha-2}}$ is α^α-competitive online primal dual algorithm for minimizing fractional flow time plus energy with power function f on a single machine.*

6.3 A Better Online Primal Dual Algorithm

The main drawback of the simple algorithm is that it does not reschedule the remaining volume of old jobs (i.e., assign them to other time slots) on the arrival of a new job. In particular, the simple algorithm does not preempt less important old jobs with the new one. We can obtain better online primal dual algorithms for this problem by rescheduling old jobs as a new job arrives.

A natural rescheduling rule is to reschedule jobs in future time slots to minimize the the fractional flow time plus energy in the remaining instance (i.e., the effective weight and volume of each old job is scaled down according to its remaining volume) assuming that there would be no future jobs. Devanur and Huang [14] called this greedy approach the conservative greedy algorithm and showed that it is $O(\alpha)$ competitive for minimizing fractional flow time plus energy.

We can further improve the conservative greedy algorithm. Note that the conservative greedy algorithm achieves the optimal fractional flow time plus energy if there are no future jobs, but is far from optimal when there are a lot of future jobs. A smarter rescheduling rule would run faster than the optimal schedule of the remaining instance to hedge between the two cases: if there are no future jobs, this approach pays more in energy comparing to the conservative greedy algorithm; if there are a lot of future jobs, however, this approach would do better than the conservative greedy algorithm because it has predicted the arrival of future jobs and has run faster in the past. Specifically, Devanur and Huang [14] considered a family of such algorithms that run C times faster than the optimal schedule of the remaining instance for some fixed parameter $C \geq 1$ and called them C-aggressive greedy algorithms (Figure 7).

Speed Scaling:	Choose speed s.t. $f^*(f'(\frac{y_t}{C}))$ equals the total weight of the remaining jobs. Let $\beta_t = \frac{1}{C} f'(\frac{y_t}{C})$ s.t. $f^*(C\beta_t)$ equals the total weight of the remaining jobs.
Job Selection:	Schedule the job with highest weight/volume ratio (highest density first).

Figure 7: C-aggressive greedy online primal dual algorithm for minimizing fractional flow time plus energy with power function f on a single machine

Moreover, Devanur and Huang [14] conducted an online primal dual analysis of C-aggressive greedy algorithms for all $C \geq 1$ and chose C to balance the case with no future jobs and the case with a lot of future jobs. By doing so, they showed the following result.

Theorem 6.3. *For polynomial power function $f(y) = y^\alpha$, the C-aggressive greedy online primal dual algorithm is $\Omega\big(\frac{\alpha}{\log \alpha}\big)$-competitive for minimizing fractional flow time plus energy on unrelated machines with $C \approx 1 + \frac{\log \alpha}{\alpha}$.*

If there are at least two machines, then Devanur and Huang [14] further showed that the above competitive ratio is asymptotically tight.

Theorem 6.4. *For polynomial power function $f(y) = y^\alpha$, there are no $o\big(\frac{\alpha}{\log \alpha}\big)$-competitive online algorithm for minimizing fractional flow time plus energy with at least two machines.*

Recall that in the previous two examples, the upper and lower bounds match exactly as they both reduce to the same differential equation. In this example, however, there is a constant gap between the upper and lower bounds. One of the reasons is that the algorithms in this example assign jobs to machines integrally, i.e., a job cannot be processed on multiple machines in parallel. We do not know whether we can derive the same form of tight upper and lower bounds as in the previous two examples if we allow parallel processing and truly focus on solving the primal convex program online.

Further, if there is only one machine, Bansal et al. [4] gave a 2-competitive online algorithm for minimizing fractional flow time plus energy with arbitrary power functions using a potential function argument. It is an interesting open question whether there is a 2-competitive online primal dual algorithm for the single machine case.

6.4 Applications

Finally, Devanur and Huang [14] also showed that the C-aggressive algorithms can be applied to minimizing (integral) weighted flow time plus energy with minor changes.

Theorem 6.5. *For polynomial power function $f(y) = y^\alpha$, there is an $\Omega\big(\frac{\alpha}{\log \alpha}\big)$-competitive online primal dual algorithm for minimizing weighted flow time plus energy on unrelated machines.*

Essentially the same algorithm and analysis can be further used to derive better competitive ratios in the resource augmentation setting, where the online algorithm can run the machines $1+\epsilon$ times faster than the offline benchmark using the same amount of energy.

Theorem 6.6. *There is a $(1+\epsilon)$-speed and $\Omega\big(\frac{1}{\epsilon}\big)$-competitive online primal dual algorithm for minimizing weighted flow time plus energy with arbitrary power functions.*

References

[1] Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the 26th Annual Symposium on Discrete Algorithms*. SIAM, 2015.

[2] S. Anand, Naveen Garg, and Amit Kumar. Resource augmentation for weighted flow-time explained by dual fitting. In *Proceedings of the 23rd Annual Symposium on Discrete Algorithms*, pages 1228–1241. SIAM, 2012.

[3] Nikhil Bansal, Niv Buchbinder, and Joseph Seffi Naor. Randomized competitive algorithms for generalized caching. In *Proceedings of the 40th Symposium on Theory of Computing*, pages 235–244. ACM, 2008.

[4] Nikhil Bansal, Ho-Leung Chan, and Kirk Pruhs. Speed scaling with an arbitrary power function. In *Proceedings of the 20th Annual symposium on discrete algorithms*, pages 693–701. SIAM, 2009.

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

[6] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK*, pages 72–87, 2003.

[7] Avrim Blum, Anupam Gupta, Yishay Mansour, and Ankit Sharma. Welfare and profit maximization with production costs. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pages 77–86. IEEE, 2011.

[8] Niv Buchbinder and Rica Gonen. Incentive compatible multi-unit combinatorial auctions: a primal dual approach. *Algorithmica (to appear).*

[9] Niv Buchbinder and Joseph 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.

[10] Niv Buchbinder and Joseph 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.

[11] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

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

[13] George B. Dantzig. *Linear programming and extensions.* Princeton University Press, 1998.

[14] Nikhil R. Devanur and Zhiyi Huang. Primal dual gives almost optimal energy efficient online algorithms. In *Proceedings of the 25th Annual Symposium on Discrete Algorithms*, pages 1123–1140. SIAM, 2014.

[15] Nikhil R. Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the 44th Annual Symposium on Theory of Computing*, pages 137–144. ACM, 2012.

[16] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the 24th Annual Symposium on Discrete Algorithms*, pages 101–107. SIAM, 2013.

[17] Anupam Gupta, Ravishankar Krishnaswamy, and Kirk Pruhs. Online primal-dual for non-linear optimization with applications to speed scaling. In *Approximation and Online Algorithms*, pages 173–186. Springer, 2013.

[18] Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: a primal dual approach. In *Proceedings of the 26th Annual Symposium on Discrete Algorithms*. SIAM, 2015.

[19] William Karush. *Minima of functions of several variables with inequalities as side constraints.* PhD thesis, Master's Thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

[20] Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, 1951.

[21] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22, 2007.

[22] Kim Thang Nguyen. Lagrangian duality in online scheduling with resource augmentation and speed scaling. In *Algorithms–ESA*, pages 755–766. Springer, 2013.