Primal-dual and dual-fitting analysis of online scheduling algorithms for generalized flow-time problems

Spyros Angelopoulos* Giorgio Lucarelli† Nguyen Kim Thang‡

1 Introduction

We consider online scheduling problems in which a set of jobs $\mathcal{J}$ arrive over time, and the jobs must be executed on a single processor. In particular, each job $j \in \mathcal{J}$ is released at time r_j and it is characterized by a *processing time* $p_j > 0$ and a *weight* $w_j > 0$, which become known after its release. The *density* of job j is $\delta_j = w_j/p_j$. Given a scheduling strategy, we denote by C_j the *completion time* of job j. The *flow time* of j is then defined as $F_j = C_j - r_j$. A natural optimization objective is to design schedules that minimize the *total weighted flow time*, i.e., $\sum_{j\in\mathcal{J}} w_j F_j$. We assume that preemptions are allowed.

Total weighted flow-time has been extensively studied. In the unweighted setting, it is well-known that the online algorithm Shortest Remaining Processing Time is optimal. In contrast, Bansal and Chan [2] showed that no algorithm is constant-competitive for minimizing total weighted flow-time on a single processor. This rather pessimistic lower bound motivated the study of the effect of *resource augmentation*, originally introduced by Kalyanasundaram and Pruhs [6]. Given some optimization objective (e.g. total flow time), an algorithm is said to be α-speed β-competitive if it is β-competitive with respect to an offline optimal scheduling algorithm of speed $\frac{1}{\alpha}$ (here $\alpha \leq 1$). In this context, Becchetti et al. [3] showed that the natural algorithm Highest-Density-First (HDF) is $(1+\epsilon)$-speed $\frac{1+\epsilon}{\epsilon}$-competitive for total weighted flow time.

Im et al. [5] introduced a generalization of the total weighted flow-time problem, in which jobs may incur non-linear contributions to the objective. More formally, they defined the *Generalized Flow-Time Problem* (GFP) in which the objective is to minimize $\sum_{j\in\mathcal{J}} w_j g(F_j)$, where $g : \mathbb{R}^+ \to \mathbb{R}^+$ is a given non-decreasing cost function with $g(0) = 0$. This extension captures many natural variants of flow-time with real-life applications; moreover, it is an appropriate formulation of the setting of optimizing simultaneously several objectives. Im et al. [5] showed that HDF is $(2+\epsilon)$-speed $O(\frac{1}{\epsilon})$-competitive algorithm for general non decreasing functions g. On the negative side, they showed that no *oblivious* algorithm is $O(1)$-competitive with speed augmentation $2-\epsilon$, for any $\epsilon > 0$; the term oblivious refers to algorithms that do not know the function g. If g is a twice-differentiable, concave function, then there is an $(1+\epsilon)$-speed $O(\frac{1}{\epsilon^2})$-competitive algorithm, while for unit size jobs and general cost functions, FIFO is $(1+\epsilon)$-speed $\frac{4}{\epsilon^2}$-competitive [5]. Convex cost functions have been studied by Fox et al. [4].

*CNRS and LIP6, University Pierre and Marie Curie, Paris, France.
†LIG, University of Grenoble-Alpes, INP, France.
‡IBISC, University of Evry, Val d'Essonne, France.

Most of the above results rely to techniques based on amortized analysis, with or without an explicit potential function. More recently, techniques based on tools from linear programming have emerged for online scheduling problems. One of the main benefits in the application of primal-dual techniques is the fact that it offers intuition on both aspects of algorithm design and analysis. The objective of this work is to present a unified framework for a class of generalized flow-time problems that is based on primal-dual and dual-fitting techniques. More precisely, we first give a primal-dual analysis of HDF for the total weighted flow time problem which, albeit significantly more complicated than the known combinatorial one, yields insights about more complex problems. We then abstract the salient ideas of this proof into a framework which is applicable to more complex objectives and uses similar intuitive geometric interpretations of the primal/dual objectives. This framework allows us to either reprove in a simpler way known results or obtain improvements as well as new results. A full version of this abstract can be found in [1].

2 Results

A common approach in obtaining a competitive scheduling algorithm is by first deriving an algorithm for the *fractional* objective. Let $q_j(t)$ be the remaining processing time of job j at time t. The fractional remaining weight $w_j(t)$ of j at time t is $w_j\frac{q_j(t)}{p_j}$. The fractional objective of the GFP problem is $\sum_j w_j(t)g'(t-r_j)$. In [4] is proved that *if an algorithm is s-speed c-competitive for online fractional GFP, then there exists an $(1+\epsilon)s$-speed $\frac{1+\epsilon}{\epsilon}c$-competitive algorithm for the integral objective, for $0<\epsilon\leq 1$.*

Let $x_j(t)\in[0,1]$ be a variable that indicates the execution rate of $j\in\mathcal{J}$ at time t. The following is a valid linear-programming formulation for fractional GFP and its dual.

$$\min \sum_{j\in\mathcal{J}} \delta_j \int_{r_j}^{\infty} g(t-r_j)x_j(t)dt \qquad \text{(P)}$$

$$\int_{r_j}^{\infty} x_j(t)dt \geq p_j \quad \forall j\in\mathcal{J} \qquad (1)$$

$$\sum_{j\in\mathcal{J}} x_j(t) \leq 1 \quad \forall t\geq 0 \qquad (2)$$

$$x_j(t)\geq 0 \quad \forall j\in\mathcal{J}, t\geq 0$$

$$\max \sum_{j\in\mathcal{J}} \lambda_j p_j - \int_0^{\infty} \gamma(t)dt \qquad \text{(D)}$$

$$\lambda_j - \gamma(t) \leq \delta_j g(t-r_j) \qquad \forall j\in\mathcal{J}, t\geq r_j \qquad (3)$$

$$\lambda_j,\gamma(t)\geq 0 \qquad \forall j\in\mathcal{J}, \forall t\geq 0$$

The primal complementary slackness (CS) condition states that for a given job j and time t, if $x_j(t)>0$, i.e., if the algorithm executes job j at time t, then it should be that $\gamma(t)=\lambda_j-\delta_j g(t-r_j)$. We would like then the dual variable $\gamma(t)$ to be such that we obtain some information about which job to schedule at time t. To this end, for any job $j\in\mathcal{J}$, we define the curve $\gamma_j(t)=\lambda_j-\delta_j g(t-r_j)$, with domain $[r_j,\infty)$, and slope $-\delta_j$. In order to ensure feasibility, our algorithm for every $t\geq 0$ choose $\gamma(t)=\max\{0,\max_{j\in\mathcal{J}:r_j\leq t}\{\gamma_j(t)\}\}$. We say that at time t the line γ_j is *dominant* if $\gamma_j(t)=\gamma(t)$. We can thus restate the primal CS condition as a *dominance* condition: if a job j is executed at time t, then γ_j must be dominant at t. Based on this, we abstract the essential properties in order to obtain optimal online algorithms for fractional objectives. We consider that the primal solution is generated by an online algorithm A. The crux is in maintaining dual variables λ_j, upon release of a new job z at time τ, such that the following properties are satisfied: ($\mathcal{P}1$) *Future dominance*: if the algorithm A executes job j at time $t\geq\tau$, then γ_j is dominant at t; ($\mathcal{P}2$) *Past dominance*: if the algorithm

A executes job j at time $t < \tau$, then γ_j remains dominant at t. In addition, the primal solution for $t < \tau$ does not change due to the release of z; and $(\mathcal{P}3)$ *Completion*: $\gamma(t) = 0$ for all $t > C_{\max}$, where $C_{\max}$ is the completion time of the last job.

Theorem 1. *Any algorithm that satisfies the properties* $(\mathcal{P}1)$, $(\mathcal{P}2)$ *and* $(\mathcal{P}3)$ *with respect to a feasible dual solution is an optimal online algorithm for the fractional GFP problem with general cost functions* g.

Corollary 2.
(i) HDF is optimal for the fractional online GFP problem with linear cost function.
(ii) HDF is optimal for the fractional online problem of minimizing $\sum_{j\in\mathcal{J}} w_j g(C_j)$.
(iii) Adapted HDF is $\max_j \frac{\max_i b_{ij}}{\min_i b_{ij}}$*-speed 1-competitive for the online Packing Scheduling Problem problem of minimizing* $\sum_{j\in\mathcal{J}} \delta_j \int_{r_j}^{\infty}(t-r_j)x_j(t)dt$ *subject to packing constraints* $\{B\mathbf{x} \leq 1, \mathbf{x} \geq 0\}$ *which must be upheld at all times, where* $B = \{b_{ij} > 0\}$.
(iv) FIFO (resp. LIFO) is optimal for the fractional online GFP problem with convex (resp. concave) costs functions and jobs of equal density.

In order to allow for competitive algorithms, we relax certain properties: $(\mathcal{Q}1)$ if the algorithm A schedules job j at time $t \geq \tau$ then $\gamma_j(t) \geq 0$ and $\lambda_j \geq \gamma_{j'}(t)$ for every other pending job j' at time t; $(\mathcal{Q}2)$ if the algorithm A schedules job j at time $t < \tau$, then $\gamma_j(t) \geq 0$ and $\lambda_j \geq \gamma_{j'}(t)$ for every other pending job j' at time t. In addition, the primal solution for $t < \tau$ is not affected by the release of z; and $(\mathcal{Q}3)$ $\gamma(t) = 0$ for all $t > C_{\max}$.

Theorem 3. *Any algorithm that satisfies the properties* $(\mathcal{Q}1)$, $(\mathcal{Q}2)$ *and* $(\mathcal{Q}3)$ *with respect to a feasible dual solution is a* $\frac{1}{1-\epsilon}$*-speed* $\frac{1}{\epsilon}$*-competitive algorithm for the fractional GFP problem with general cost functions* g.

Corollary 4. *FIFO (resp. HDF) is* $\frac{1}{1-\epsilon}$*-speed* $\frac{1}{\epsilon}$*-competitive for the fractional online GFP problem with general cost functions and equal-density jobs (resp. for the fractional online GFP problem with concave cost functions).*

References

[1] S. Angelopoulos, G. Lucarelli, and N. K. Thang. Primal-dual and dual-fitting analysis of online scheduling algorithms for generalized flow-time problems. *CoRR*, arXiv:1502.03946, 2015.

[2] N. Bansal and H.-L. Chan. Weighted flow time does not admit $o(1)$-competitive algorithms. In *SODA*, pages 1238–1244, 2009.

[3] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, and K. Pruhs. Online weighted flow time and deadline scheduling. *J. Discrete Algorithms*, 4(3):339–352, 2006.

[4] K. Fox, S. Im, J. Kulkarni, and B. Moseley. Online non-clairvoyant scheduling to simultaneously minimize all convex functions. In *APPROX-RANDOM*, pages 142–157, 2013.

[5] S. Im, B. Moseley, and K. Pruhs. Online scheduling with general cost functions. *SIAM Journal on Computing*, 43(1):126–143, 2014.

[6] B. Kalyanasundaram and K. Pruhs. Speed is as powerful as clairvoyance. *J. ACM*, 47(4):617–643, 2000.