The Complexity of Scheduling for $p$-Norms of Flow and Stretch
(Extended Abstract)

Benjamin Moseley$^1$, Kirk Pruhs$^{2,**}$, and Cliff Stein$^3,**$

$^1$ Toyota Technological Institute, Chicago IL, 60637, USA
moseley@ttic.edu

$^2$ Computer Science Department, University of Pittsburgh, Pittsburgh, PA 15260, USA
kirk@cs.pitt.edu

$^3$ Department of Industrial Engineering & Operations Research, Columbia University,
Mudd 326, 500W 120th Street, New York, NY 10027, USA
cliff@ieor.columbia.edu

Abstract. We consider computing optimal $k$-norm preemptive schedules of jobs that arrive over time. In particular, we show that computing the optimal $k$-norm of flow schedule, $1 \mid r_j, pmtn \mid \sum_j (C_j - r_j)^k$ in standard 3-field scheduling notation, is strongly NP-hard for $k \in (0, 1)$ and integers $k \in (1, \infty)$. Further we show that computing the optimal $k$-norm of stretch schedule, $1 \mid r_j, pmtn \mid \sum_j ((C_j - r_j)/p_j)^k$ in standard 3-field scheduling notation, is strongly NP-hard for $k \in (0, 1)$ and integers $k \in \cup (1, \infty)$.

1 Introduction

In the ubiquitous client-server computing model, multiple clients issue requests over time, and a request specifies a job for the server to perform. When the requested jobs have widely varying processing times — as is the case for compute servers, database servers, web servers, etc. — the server system generally must allow (presumably long) jobs to be preempted for waiting (presumably smaller) jobs in order to provide a reasonable quality of service to the clients. The most commonly considered and most natural quality of service measure for a job $j$ is the flow/waiting/response time, which is $C_j - r_j$ the duration of time between time $r_j$ when the request is issued, and time $C_j$ when the job is completed. Another commonly considered and natural quality of service measure for a job $j$ is the stretch/slowdown, $(C_j - r_j)/p_j$, the flow time divided by the processing time requirement $p_j$ of the job. The stretch of a job measures how much time the job took relative to how long the job would have taken on a dedicated server. Flow time is probably more appropriate when the client has little idea of the time required for this requested job, as might be the case for a non-expert database client. Stretch is probably more appropriate when the client has at least an approximate idea of the time required for the job, as would be the case when clients are requesting static content from

* Supported in part by NSF grants CCF-0830558, CCF-1115575, CNS-1253218, and an IBM Faculty Award.

** Research partially supported by NSF grant CCF-0915681.
a web server (e.g. when requesting large video files clients will expect/tolerate a longer
response than requesting small text files).

The server must have some scheduling policy to determine which requests to prior-
itize in the case that there are multiple outstanding requests. To measure the quality of
service of the schedule produced by the server’s scheduling policy, one needs to com-
hine the quality of service measures of the individual requests. In the computer systems
literature, the most commonly considered quality of service measure for a schedule is
the 1-norm, or equivalently average or total, of the quality of service provided to the
individual jobs. Despite the widespread use, one often sees the concern expressed that
average flow is not the ideal quality of service measure in that an optimal average flow
schedule may “unfairly starve” some longer jobs. Commonly what is desired is a qual-
ity of service measure that “balances” the competing priorities of optimizing average
quality of service and maintaining fairness among jobs. The mathematically most nat-
ural way to achieve this balance would be to use the 2-norm (or more generally the
$k$-norm for some small integer $k$).

The $k$-norms of flow time and stretch have been studied in the scheduling the-
ory literature in a variety of settings: on a single machine [BP10b] [BP10a], mul-
tiple machines [CGKK04, BT06, FMTI, MITI, AGK12], in broadcast scheduling
[EIM11, CIM09, GIK+10], for parallel processors [EIM11, GIK+10] and on speed
scalable processors [GKP12]. The choice of $k$ depends on the desired balance of aver-
age performance with fairness. For example, the 2-norm is used in the standard least-
squares approach to linear regression, but the 3-norm is used within $\LaTeX$ to determine
the best line breaks. Conceivably there are also situations in which one may want to
choose $k < 1$, say when a client wants a job to be completed quickly, but if the job
is not completed quickly, the client does not care so much about how long the job is
delayed.

**Directly Related Previous Results:** In what is essentially folklore, the following is
known for optimizing a norm of flow time offline with release dates and preemption:

- the optimal 1-norm schedule can be computed in polynomial time by the greedy
  algorithm Shortest Remaining Processing Time (SRPT), and
- the optimal schedule for the $\infty$-norm, of either flow and stretch, can be computed
  in polynomial time by combining a binary search over the maximum flow or stretch
  and the Earliest Deadline First (EDF) scheduling algorithm, which produces a dead-
  line feasible schedule if one exists.

Surprisingly, despite the interest the in $k$-norms of flow time and stretch, the complexity
of computing an optimal $k$-norm of flow schedule, for $k \neq 1$ or $\infty$, and the complexity
of computing an optimal $k$-norm of stretch schedule, for any $k$, were all open.

### 1.1 Our Results

We show that for all integers $k \geq 2$, and for all $k \in (0, 1)$, the problem of finding a
schedule that minimizes the $k$-norm of flow is strongly NP-hard. Similarly, we show
that for all integer $k \geq 2$, and for all $k \in (0, 1)$, the problem of finding a schedule
that minimizes the $k$-norm of stretch is strongly NP-hard. This rules out the existence