Fairness and Scheduling in Single Server Queues

Adam Wierman

Computing and Mathematical Sciences Department
California Institute of Technology
Pasadena, California, USA 91125

Abstract

Traditionally, the study of scheduling policies has focused on performance metrics such as response time, queue length, and throughput. However, the more vague notion of 'fairness' is often equally or more important than these traditional performance metrics. But, the concept of fairness is difficult to define and so it has been studied only sporadically. This has changed over the last decade and a growing literature providing an analytic framework for studying fairness has emerged. This article surveys recent developments, which include a rich variety of fairness metrics as well as a growing understanding of the fairness of common scheduling policies.

Keywords: Queueing theory, Scheduling theory, Fairness

1. Introduction

Traditionally, the study of scheduling policies has focused on performance metrics such as the response time (sojourn time, flow time), queue length, throughput, etc. These metrics have broad importance in computer systems, manufacturing systems, communication systems, etc. However, often the more vague notion of 'fairness' is as important or even more important than optimizing traditional performance measures. For example, in addition to guaranteeing that the mean response time across all customers is small, it is often important to ensure that individual customers/jobs are treated 'fairly'. But, the concept of 'fairness' is amorphous and difficult to define, in large part because it can have a very different meaning depending on the setting considered. Resultantly, there is far less work studying the 'fairness' of scheduling policies than there is studying the 'performance' of scheduling policies. However, in the last decade, there has been a new focus on fairness emerging and the goal of this article is to provide an introduction to this growing literature.

An emerging focus on fairness

The motivation for this survey is the recent focus on fairness that has emerged within the computer systems community, which stems primarily from the following issue: When optimizing traditional performance measures such

as mean response time, it is well-known that designs should give priority to jobs with small sizes at the expense of jobs with large sizes. For example, Shortest Remaining Processing Time (SRPT) is well-known to minimize mean response time across general arrival processes and job size distributions [47] and, among non-preemptive policies, Shortest Job First (SJF) optimizes the mean response time [11]. As a result, designs that give priority to jobs with small sizes have been suggested for a variety of computer systems in recent years, such as [20, 38, 36, 37, 22, 56, 14]. However, the adoption of these new designs has been slowed by concerns about the unfairness of these policies. Specifically, it is natural to worry that large job sizes may be 'starved' of service under a policy that gives priority to small job sizes, which would result in large job sizes having response times that are unfairly long and variable [6, 48, 49, 50].

These worries have recurred nearly everywhere size based policies have been suggested. Consider, for example, the case of web servers, where recent designs have illustrated that giving priority to requests for small files can significantly reduce response times [20, 38]. However, it is important that this improvement does not come at the expense of providing large job sizes unfairly large response times, which are typically associated with the important requests. For example, at an online shopping site the large requests are often the shopping cart transactions. The same tradeoff has appeared across diverse computer system application areas, e.g., process scheduling [14], routers [36, 37], wireless networks [22], transport protocols [56], and beyond.

To address worries such as those described above, the research community has sought over the last decade to develop a theoretical framework for studying the fairness of scheduling policies. Research studying the fairness of scheduling policies can largely be divided into two pieces. First, a large literature has focused on developing fairness measures and studying the fairness of policies with respect to jobs in the context of a single resource (a single server queue), e.g., [40, 39, 45, 51, 53, 52]. Second, a large (and nearly distinct) literature has focused on developing fairness measures and studying the fairness of policies with respect to flows in the context of multiple resource environments, e.g., sharing network resources among flows, e.g., [24, 12, 27, 32, 7, 28]. Due to the breadth of each of these two literatures it is necessary to focus on only one in this article, so we discuss only recent results studying fairness across jobs in a single server queue, though we provide context and pointers to the multi-resource literature where relevant. As we will see, even in the 'simple' case of fairness across jobs in a single resource setting there are a wide variety of interpretations of the meaning of 'fair' and so focusing on this setting is illustrative.

Whenever definitions of fairness are discussed, there are an unending array of philosophic, social, and psychological issues that are immediately brought into consideration, which is precisely the difficulty in arriving at a unified, widely-accepted definition. This survey attempts to take a pragmatic view of fairness, motivated by the recent focus in computer systems, and thus describes the motivation for each of the metrics presented without delving deeply into the social, philosophical, and psychological aspects. However, the reader who is interested in these issues can find a number of interesting works that specifically

attempt to provide insight into these areas including [35, 29, 19] and the recent survey of Avi-Itzhak et al. [4], which does a thorough job of discussing such issues for many of the metrics mentioned in this survey.

Defining fairness in a single server queue

As we have mentioned, fairness is an amorphous concept that is nearly impossible to define in a universal way. This fact has proven to be a major stumbling block for analytic study because the notion of fairness in one application may be very different than in another, and often, even within the context of a single application there can be many conflicting views of what is meant by 'fair.'

To highlight the fact that fairness is very application-dependent, notice that if the queue in question has a limited resource that is being distributed (e.g. a ticket box office), then it is more fair to serve jobs in the order they arrive. However, in many other settings (e.g. a grocery store or a bank) it is quite acceptable to allow small jobs to bypass large jobs. In contrast, if the setting is a hospital, things change completely: it is more fair to serve the more urgent job, regardless of the job sizes or arrival order.

To further illustrate the subtlety in defining the fairness of scheduling policies, consider the following two simple examples:

(i) Suppose jobs a and b are the same size, and a enters the queue slightly before b.

(ii) Suppose jobs c and d are very large and very small respectively, and job c enters the queue slightly before job d.

Most people agree that (in most settings) it is fair to serve job a before job b and to serve job d before job c. Specifically, there is a consensus that in most cases it is 'unfair' for small jobs to queue behind larger ones and that it is 'unfair' for a job that arrived later to bypass jobs that arrived earlier. We refer to these two intuitive notions of fairness as 'proportional fairness'[1] and 'temporal fairness' respectively. Proportional fairness refers to the idea that it is 'fair' for the response time of jobs to be proportional to the job sizes, and temporal fairness refers to the idea that it is 'fair' to serve jobs in the order of arrival. Note that, though both of these notions of fairness are intuitive, they are clearly competing ideas – when a small job arrives some length of time after a large job it is unclear which job it is more 'fair' to serve first. In fact, when considering temporal fairness First Come First Served (FCFS) is the most fair policy, but FCFS is clearly unfair to small jobs when proportional fairness is considered. In contrast, Processor Sharing (PS) scheduling, which shares the server evenly among all the jobs in the system, is intuitively fair and provides both temporal fairness and proportional fairness for the two particular examples

[1]The reader should be careful not to confuse this notion of 'proportional fairness' with the one introduced by Kelly et al. in [24] that is commonly used in the network bandwidth sharing context.

above. We chose FCFS and PS for examples here because it turns out that in many proposed fairness metrics either PS or FCFS is the fairness ideal.

It is important to point out that both of these notions of fairness are appropriate, to varying degrees, in a variety of applications. Even if we limit the discussion to the domain of computer systems, we can find examples where each is important. For example, proportional fairness has recently received wide attention in the computer systems community due to design proposals that give priority to small jobs at the expense of large jobs, e.g., in [20, 38, 36]. In each of these cases, the worry is whether large jobs are unfairly starved of service as a result of the priority given to small jobs. On the other hand, temporal fairness is important in computer applications such as video streaming, where out-of-order delivery of frames leads to jitter in the video stream.

The notions of 'proportional fairness' and 'temporal fairness' dominate the literature studying fairness in the single resource environment, and the remainder of this article is structured so as to contrast the metrics and results across these two notions. In particular, we first focus on proportional fairness metrics (Section 3), then move to a discussion of temporal fairness metrics (Section 4), and finally discuss approaches for developing fairness metrics that combine aspects of both proportional and temporal fairness (Section 5). A running theme throughout all discussions will be the seeming conflict between 'fairness' and 'performance' (as measured by mean response time) as well as the conflict between maintaining proportional fairness and temporal fairness.

2. Preliminaries

In this article, we limit our discussion of fairness to the single server queue. We will mostly focus on the GI/GI/1 setting, though occasionally results will be limited to either the M/GI/1 or GI/D/1 setting. Throughout, we will denote the mean arrival rate of jobs by λ and a generic job size by the random variable X having p.d.f. $f(x)$, c.d.f. $F(x)$, and c.c.d.f. $\overline{F}(x) = 1 - F(x)$. Additionally, we define the load $\rho = \lambda E[X]$ and limit consideration to systems with $0 < \rho < 1$. We denote the length of a generic busy period by B. Occasionally, we will consider the load made up by jobs of size $< x$, denoted by $\rho(x) = \lambda \int_0^x t f(t) dt$.

Our focus will typically be on the response time (sojourn time, flow time) of jobs, which is defined as the time between the arrival and departure of a job. We denote a generic response time by T and the conditional response time for a job of size x by $T(x)$. We use $E[T(x)]^P$ to denote the expected conditional response time under policy P. When describing the asymptotic behavior of $T(x)$ we will use the notation $f(x) = \Theta(g(x))$ as $x \to a$ to denote $0 < \liminf_{x \to a} |f(x)/g(x)| \leq \limsup_{x \to a} |f(x)/g(x)| < \infty$.

We will be discussing a wide array of scheduling policies in order to highlight the results proven using each of the various fairness metrics we discuss. To provide easy reference to the acronyms and definitions of the scheduling policies, we summarize them in Table 1. Table 1 includes a variety of policies from simple common policies like FCFS to more complicated priority based policies like SRPT and PSJF. However, all of the policies in Table 1 are *work-conserving*, i.e.,

FB	Foreground-Background preemptively serves those jobs that have the smallest age (attained service). FB is known by a variety of other names including Least Attained Service (LAS) and Shortest Elapsed Time first (SET).
FCFS	First Come First Served serves jobs in the order they arrive. FCFS is also commonly referred to as First In First Out (FIFO).
LCFS	Last Come First Served non-preemptively serves the job that arrived the most recently. LCFS is also commonly referred to as Last In First Out (LIFO).
LRPT	Longest Remaining Processing Time preemptively serves the job in the system with the largest remaining processing time.
LJF	Longest Job First non-preemptively serves the job in the system with the largest original size.
PLCFS	Preemptive Last Come First Served preemptively serves the most recent arrival.
PLJF	Preemptive Longest Job First preemptively serves the job in the system with the largest original size.
PS	Processor Sharing serves all customers simultaneously, at the same rate.
PSJF	Preemptive Shortest Job First preemptively serves the job in the system with the smallest original size.
ROS	Random Order of Service non-preemptively serves a job that is chosen uniformly at random from the queue.
SJF	Shortest Job First non-preemptively serves the job in the system with the smallest original size.
SRPT	Shortest Remaining Processing Time preemptively serves the job with the shortest remaining size.

Table 1: *A brief description of the scheduling policies discussed in this paper.*

they serve with the full service rate whenever there is work in the queue, and we will limit the discussion in the paper to such policies. For an introduction and survey of results about the policies in Table 1 see [11, 26]. In addition to considering the individual policies in Table 1, we discuss results about scheduling 'classifications', which are groups of scheduling policies defined by the technique used. For example, the class of SMART policies [34, 55] includes policies that give priority to small jobs and the class of FOOLISH policies [51, 52] includes policies that give priority to large jobs. Other classifications are more easily understood from their names, such as preemptive size based policies and age based policies, which preemptively assign priority based on the size or age (attained service) of jobs. For formal definitions of these scheduling classifications and a complete survey of recent analyses we refer the reader to [51, 52].

3. Proportional fairness metrics

We start by discussing fairness metrics that focus on 'proportional fairness'. These metrics capture the intuitive notion that it is fair for large jobs to have large response times and for small jobs to have small response times. Philosophically, one can view proportional fairness as related to an Aristotelian notion of fairness: like cases should be treated alike; different cases should be treated differently; and different cases should be treated differently in proportion to the difference at stake [42].

The issue of proportional fairness came to attention recently due to the work of Harchol-Balter et al. which proposed using a design based on SRPT for web servers and then sought to dispel worries about unfairness toward large job sizes by studying the fairness or 'starvation' of SRPT [20]. In this work, the goal was to dispel the feeling that SRPT 'starves' large jobs of service (relative to small jobs). From this initial focus on SRPT a large literature developed, which now characterizes the proportional fairness of nearly all common policies.

It is important to remember that the measures discussed in this section isolate proportional fairness from the issue of temporal fairness, which is discussed in Section 4. Section 5 will then discuss metrics that combine the two notions of fairness.

3.1. Proportional fairness in expectation

The first definition of fairness we discuss is the definition that began the recent focus on fairness in the computer systems community. This definition of proportional fairness *in expectation* was used in [5] to discuss the fairness of a new design for web servers based on SRPT, and then was generalized by Wierman & Harchol-Balter in [53] as follows:

Definition 1. *Let $0 < \rho < 1$ in an M/GI/1. A job size x is treated **fairly** under policy P, service distribution X and load ρ if*

$$\frac{E[T(x)]^P}{x} \leq \frac{1}{1-\rho}$$

*Otherwise a job size x is treated **unfairly**. A scheduling policy P is **fair** if every job size is treated fairly. Otherwise P is **unfair**.*

There are a few things that are important to notice about this definition. First, note that this definition considers fairness of particular *job sizes* rather than of individual jobs. This is due to the fact that it was developed initially in order to contrast the experience of small jobs and large jobs under SRPT. However, in general, this is a natural level of specificity for a measure of proportional fairness since this notion of fairness is very size-based. A second point to note is that Definition 1 actually consists of two pieces: a *metric*, $E[T(x)]/x$, that measures the fairness experienced by a *job size*, and a *criterion*, $1/(1-\rho)$, which determines if a *policy* is fair. Each of these pieces can be motivated in a variety of ways, which we describe below.

Let us first discuss some justifications for the metric $E[T(x)]/x$ for determining the fairness to the class of jobs with size x. The metric $E[T(x)]/x$, which is often referred to as the *mean conditional slowdown*, intuitively aligns with the spirit of proportional fairness. More formally, when comparing $E[T(x)]^P$ across x, we want a metric that scales $E[T(x)]^P$ appropriately to allow for (non-trivial) comparison of $E[T(x)]^P$ between small and large x. For $E[T(x)]^P$, $1/x$ is an appropriate scaling factor because $E[T(x)]^P = \Theta(x)$ under all work conserving scheduling policies [21], and thus we need to normalize by $1/x$ to provide non-trivial comparisons between small and large job sizes. Note that slowdown is also used as a measure of fairness in a variety of other, earlier, works, e.g. [26, 6].

To justify the criterion $1/(1-\rho)$ for determining whether a policy is fair, we again start with an intuitive justification: PS is typically thought of as a fair policy because at every instant every job in the system receives an equal share of the server. Further, PS satisfies the idea that $E[T(x)]$ should be proportional to x, since $E[T(x)]^{PS}/x = 1/(1-\rho)$. Thus, $1/(1-\rho)$ is a natural choice for a fairness criterion. In addition, it is a practical choice because in many computer applications PS is the status quo, and by comparing with the status quo we develop an understanding of how new designs will affect response times. More formally, we can also justify the criterion $1/(1-\rho)$ by realizing that it represents a form of min-max fairness (a.k.a. Pareto efficiency). In particular, Wierman & Harchol-Balter [53] prove that $\min_P \sup_x E[T(x)]^P/x = 1/(1-\rho)$, which highlights that a policy is being unnecessarily unfair to a job of size x if it has $E[T(x)]/x > 1/(1-\rho)$. Note that this idea of min-max fairness is commonly applied in computer systems and has been used, for example, in the context of bandwidth sharing in networks [28, 31]. Finally, it is important to point out that in addition to the above justifications, $1/(1-\rho)$ turns out to be a *useful* criterion because it distinguishes between distinct patterns of behavior of policies with respect to $E[T(x)]^P/x$ (see Figure 1).

In addition to the justifications described above, there is also a very recent piece of work of Doroudi & Wierman [13], which provides an economic justification for Definition 1. Specifically, [13] considers a setting where jobs of size x receive reward $R(x)$ from completing service and pay cost C per second while in the system. Further, jobs of size x make strategic decisions about whether to join a queue or not based on only knowing $E[T(x)]$. It turns out that in the case of linear $R(x)$, a policy is fair under Definition 1 if and only if it ensures that the equilibria behavior of job sizes is for all to join the queue with an equal probability. Thus, Definition 1 ensures that *all* job sizes make the same strategic decision about whether or not to enter the queue, which highlights that no job size feels discriminated against.

We have discussed a number of justifications for Definition 1, however it should be highlighted that Definition 1 focuses completely on proportional fairness and ignores the issue of temporal fairness entirely. Further, Definition 1 is defined only in the context of the M/GI/1 queue and may be difficult, or impossible to extend to more general settings such as multi-queue systems or network settings. However, a key benefit of Definition 1 is its simplicity, which

has allowed the analysis of nearly all common scheduling policies.

It turns out that it is useful to classify scheduling policies based on whether they (i) treat all job sizes fairly or (ii) treat some job sizes unfairly. Curiously, some policies may fall into either type (i) or type (ii) depending on the load and service distribution. Specifically, [53] defines *three types of unfairness*:

Definition 2. *Let $0 < \rho < 1$ in an M/GI/1 queue. A scheduling policy P is: (i) **Always Fair** if P is fair for all such ρ and X; (ii) **Sometimes Fair** if P is fair under some ρ and X and unfair under other ρ and X; or (iii) **Always Unfair** if P is unfair for all ρ and X.*

To develop an understanding of Definitions 1 and 2, let us start to discuss a few specific policies. First, it is easy to see that PS and PLCFS are Always Fair because

$$E[T(x)]^{PLCFS} = E[T(x)]^{PS} = \frac{x}{1-\rho}$$

Further, PS and PLCFS are two examples of a larger class of 'symmetric' policies [23], which all have the same $E[T(x)]$. Thus, there are a number of policies that are fair under all service distributions and all loads.

However, most common policies are not Always Fair. In fact, most are Always Unfair. For example, FCFS is Always Unfair – the smallest job size in the service distribution is treated unfairly [53]. But, somewhat surprisingly, it turns out that SRPT is not Always Unfair, it is Sometimes Fair. In particular, the fairness of SRPT was first studied by Bansal & Harchol-Balter [5] under the assumption that $E[X^2] < \infty$. These initial results were later extended by Wierman & Harchol-Balter [53], and then by Brown [10], who was the first to consider fairness when $E[X^2] = \infty$. We summarize the major results in the following theorem:

Theorem 1. *SRPT is Sometimes Fair. SRPT is fair when $\rho \leq 0.5$ or when the service distribution is regularly varying[2] with rate $\alpha \in (1, 1.5)$. However, when $E[X^2] < \infty$, under all service distributions there exists a $\rho_c < 1$ such that for all $\rho > \rho_c$ SRPT is unfair.*

Additionally, even in the settings where SRPT is unfair, the degree of unfairness to large job sizes is not as bad as one might expect.

Theorem 2. *All x such that $\rho(x) \leq \max(1/2, 2\rho/3, 1 - \sqrt{1-\rho})$ are treated fairly under SRPT. Further, for any x that is treated unfairly,*

$$\frac{E[T(x)]^{SRPT}}{x} \leq \left(\frac{2-\rho}{2(1-\rho)}\right)\frac{1}{1-\rho}.$$

[2]Regularly varying distributions are a class of distributions that have the same tail behavior as the Pareto distribution. Formally, a service distribution is regularly varying if $\overline{F}(x) = L(x)x^{-\alpha}$ for some slowly varying $L(x)$, i.e., $L(x)$ such that for all $y > 0$, $L(yx)/L(x) \to 1$ as $x \to \infty$.

Notice that only a small fraction of jobs are treated unfairly and that these jobs experience a small degree of unfairness. Further, it is important to note that as ρ increases, so does the lower bound $1 - \sqrt{1-\rho}$ on $\rho(x)$. In fact, this bound converges to 1 as $\rho \to 1$, which signifies that the size of the smallest job that might be treated unfairly is increasing unboundedly as ρ increases.

Note though that although we have focused the above discussion on SRPT, a wide variety of other common policies have been analyzed with respect to Definition 1 in [5, 10, 16, 21, 36, 37, 53]. The reader may find illustrations of the fairness behavior of FB, PSJF, and many other common policies in Figures 1 and 2. In fact, Figure 2 highlights that not only have individual policies been analyzed, but scheduling classifications have also been analyzed.

One thing that Figure 2 exposes is that there are very few common policies that are Always Fair, and further, those policies that are Always Fair have mean response times that can be far from optimal (recall that all symmetric policies have $E[T(x)] = x/(1-\rho)$). In fact, Figure 2 presents a discouraging picture: no common scheduling heuristics or techniques can result in policies that are both Always Fair and near-optimal for mean response time.

The search for a fair policy with near optimal performance proved elusive until Friedman and Henderson presented a new policy called Fair-Sojourn-Protocol (FSP), which provides the first example of a fair policy that significantly improves upon the performance of PS [15].

FSP works as follows. It maintains a virtual PS queue, with the same arrival process, and schedules at any point in time the job that will be completed first in the virtual PS queue. Thus, FSP can be thought of as performing SRPT on the remaining times of a virtual PS system. Given this definition, it is almost immediate to conclude that FSP is Always Fair; however, a detailed performance analysis of FSP has proven elusive. Thus, in order to understand how much FSP improves response times over PS, researchers have been limited to simulation studies such as those in [15, 17, 18]. These studies have shown that FSP provides significantly improved mean response time compared to PS, but that in high load it does not match the optimal mean response time of SRPT. An important open question is to better understand the performance of FSP.

3.2. Beyond proportional fairness in expectation

To this point, we have discussed proportional fairness only *in expectation*. However, worries about the fairness are not limited to the mean. People worry that large jobs will experience both larger and *more variable* response times [6, 48, 49, 50]. Thus, it is important to develop a framework for comparing distributional properties of $T(x)$ across x, not just $E[T(x)]$.

To start our discussion of distributional behavior of proportional fairness we first focus only on the experience of *large job sizes*. By focusing on only the behavior of large job sizes, we highlight the foundation for how Definition 1 is generalized beyond proportional fairness in expectation. Also, a study of large job sizes is of practical importance because it is worries about the performance

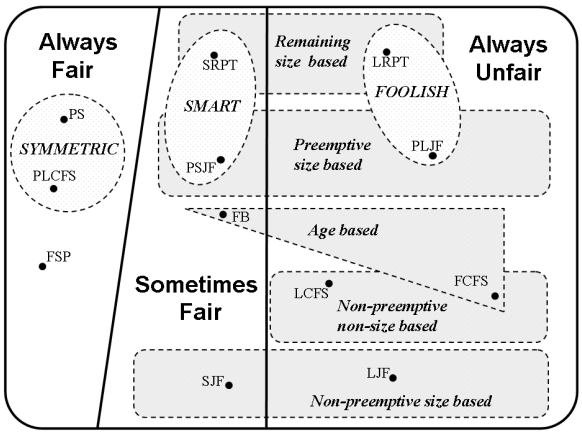


Figure 2: An illustration of the classification of common scheduling policies and prioritization techniques with respect to Definition 2 of proportional fairness.

and are calculated using the logarithm of the moment generating function, i.e., $\log E[e^{tX}] = \sum_{i=1}^{\infty} \kappa_i[X] \frac{t^i}{i!}$. In the following, let $1\{C\}$ be the indicator function for condition C.

Theorem 4. *In an M/GI/1 queue with $E[X^{i+1}] < \infty$,*

$$\lim_{x \to \infty} \frac{\kappa_i[T(x)]}{x} \leq 1\{i = 1\} + \lambda E[B^i]$$

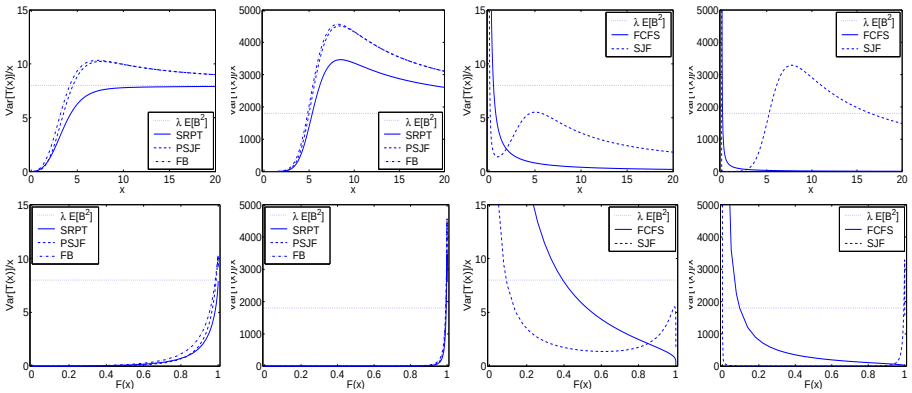
under all work conserving policies and equality holds for $P \in \{PS,\ PLCFS,\ SRPT,\ FB,\ SMART\}$ even when $E[X^{i+1}] = \infty$ as long as $E[X^i] < \infty$. If the policy is also non-preemptive, then $\lim_{x\to\infty} \kappa_i[T(x)]/x = 1\{i = 1\}$.

Theorem 4 provides a 'central limit theorem scaling' in the case of $i = 2$ (since $\kappa_2[T(x)] = Var[T(x)]$) as opposed to the law of large numbers scaling in Theorem 3. Further, larger i capture more detailed information about the distribution of $T(x)$. It is worth pointing out that though the $1\{i = 1\}$ may appear strange at first, it is a fundamental consequence of the fact that the first cumulant is shift-equivariant while all others are shift-invariant, i.e., letting c be a constant, $\kappa_1[Y + c] = \kappa_1[Y] + c$ and for $i \geq 2$, $\kappa_i[Y + c] = \kappa_i[Y]$.

Theorem 4 then motivates the following generalization of Definition 1 to provide a measure of proportional fairness in distribution, introduced by Wierman & Harchol-Balter in [54]:

Definition 3. *Let $0 < \rho < 1$ and $E[X^i] < \infty$ in an M/GI/1. A job size x is treated **fairly** under policy P, service distribution X, and load ρ if for all i,*

$$\frac{\kappa_i[T(x)]^P}{x} \leq 1\{i = 1\} + \lambda E[B^i].$$



(a) Low load ($\rho = 0.5$) (b) High load ($\rho = 0.9$) (c) Low load ($\rho = 0.5$) (d) High load ($\rho = 0.9$)

Figure 3: The conditional variance of response time is illustrated under a variety of both preemptive and non-preemptive policies. The service distribution is exponential with mean 1. The dotted line shows the criterion for fairness.

Otherwise a job size x is treated ***unfairly****. A scheduling policy P is* ***fair*** *if every job size is treated fairly. Otherwise P is* ***unfair****.*

There are many parallels between Definition 1 and Definition 3. First note that the metric and criterion in Definition 1 match the metric and criterion for $i = 1$ in Definition 3: $\kappa_1[T(x)]/x = E[T(x)]/x$ and $1\{i = 1\} + \lambda E[B] = 1 + \rho/(1-\rho) = 1/(1-\rho)$. Further, the motivation for the metric and criterion parallel the motivation for the metric $E[T(x)]/x$ and the criterion $1/(1-\rho)$ in many ways.

In both Definition 1 and Definition 3, the *metric* is motivated by the growth rate of the cumulant moments of $T(x)$ as $x \to \infty$. Specifically, the metric must scale the cumulant moments of $T(x)$ appropriately to allow for comparison of moments of $T(x)$ between small and large x. For $E[T(x)]^P$, it is clear that $1/x$ is an appropriate scaling factor because $E[T(x)]^P = \Theta(x)$ under all work conserving scheduling policies. For higher cumulant moments of $T(x)$, the correct scaling factor is not obvious; however Theorem 4 illustrates that $\kappa_i[T(x)]$ is $\Theta(x)$ for common preemptive policies and $O(x)$ for all work conserving policies. Hence, scaling by $1/x$ is appropriate.

The motivation for the criteria in Definition 3 again parallels that for the criterion in Definition 1; however it is not as cut-and-dry. It is not known whether $1\{i = 1\} + \lambda E[B^i]$ provides a min-max notion of fairness for $i > 1$, nor is it known whether the economic justification can be extended to higher cumulant moments. But, there are many parallels between $1\{i = 1\} + \lambda E[B^i]$ and $1/(1-\rho)$. For instance, Theorem 4 illustrates that the criterion in Definition 3 also serves as the limit for $\kappa_i[T(x)]/x$ under many common scheduling policies just as $1/(1-\rho)$ does for $E[T(x)]/x$. Further, Wierman & Harchol-Balter [54]

prove that in the case of $i = 2$, when $\kappa_2[T(x)]/x = Var[T(x)]/x$, the criterion $\lambda E[B^2]$ does indeed differentiate between contrasting $Var[T(x)]/x$ behaviors and that the behaviors in $Var[T(x)]/x$ parallel those for $E[T(x)]/x$. This is illustrated in Figure 3. However, an important open question that remains is to show that $\kappa_i[T(x)]$ exhibits parallel behavior for $i > 2$.

4. Temporal fairness metrics

The prior section considers *proportional fairness*, which captures the idea that it is fair for a job's response time to be proportional to its job size. In this section, we focus on a second notion of fairness – *temporal fairness* – which captures the idea that it is fair for jobs to be served in the order they arrive, i.e., to serve jobs according to *seniority*.

Like proportional fairness, temporal fairness is natural in many settings. For example, in computer systems applications such as scheduling flows in routers, there is a tension between providing flows small overall mean response times, which requires prioritizing short flows, and ensuring that streaming flows (which tend to be large) do not experience jitter [36, 37].

The tradeoff between traditional performance metrics (e.g. mean response time) and temporal fairness is perhaps best illustrated by the fact that the only *strictly* temporally fair policy is FCFS, which serves jobs in the order they arrive; however, FCFS can have extremely large mean response times under highly variable service distributions. Thus, in designing a policy for an application like web servers and routers where both temporal fairness and mean response times are important, one must strike a balance between fairness and 'performance'.

In this section we highlight two recent measures of temporal fairness that have emerged in the literature: 'order fairness' and 'politeness'. The interested reader can find other examples of temporal fairness measures discussed in [19].

One important thread that we discuss in this section is the conflict between temporal fairness and proportional fairness. It is intuitively clear that the most fair policies with respect to temporal fairness is FCFS and we have already discussed that FCFS is unfair with respect to proportional fairness. It turns out the reverse is also true, policies that are fair in the sense of proportional fairness are unfair in the sense of temporal fairness.

4.1. Order fairness

The first temporal fairness metric that we discuss is the one that began the recent focus on temporal fairness. The measure, termed 'order fairness', was introduced by Avi-Itzhak & Levy in [3] and is defined as follows:

Definition 4. *Consider a GI/GI/1 queue and a non-preemptive policy P, the* ***expected order fairness*** *is defined as $E[OF]^P = Var[W]^P$, where W is the stationary waiting time.*

When interpreting this measure, smaller $E[OF]$ implies that the policy is more fair. Note that the use of low variance as an indicator of fairness is quite

intuitive and can be traced back to a number of sources including [26, 25]. More recently, [30] also uses variance to measure fairness.

To justify Definition 4, the authors of [3] give a novel axiomatic derivation. To explain this derivation let us first limit the discussion to the GI/D/1 non-preemptive setting. In this setting, [3] justifies Definition 4 by presenting four axioms that any fairness metric should satisfy and then characterizes completely the set of metrics that satisfy these axioms.

Note that in the GI/D/1 non-preemptive queue, a scheduling policy is defined by an ordering of the queue. The four axioms presented in [3] each characterize how the fairness measure should react when two jobs are interchanged by the scheduling policy, i.e., when the positions of jobs i and j in the service order are switched. Define $f_{i,j,P}$ to be the change in the fairness measure after switching the positions of i and j under policy P and let P' be the resulting policy. Then, the four axioms presented by [3] are:

- *Monotonicity under job interchange*: $f_{i,j,P}$ is strictly increasing in the seniority difference of jobs i and j.
- *Reversibility under job interchange*: $f_{i,j,P} = -f_{j,i,P'}$.
- *Independence of position and time*: $f_{i,j,P}$ is independent of the positions of i and j in the queue and of the time when the switch occurs.
- *Independence of customers not interchanged*: $f_{i,j,P}$ is independent of all customers in the queue other than i and j.

One of the major results in [3] is to prove that any fairness measure that satisfies these axioms must be of the form:

$$c \sum_i a_i d_i^P + \alpha_i \tag{4.1}$$

where $c > 0$ and α are constants and d_i^P is the displacement of the ith job under P (i.e. the number of positions job i is pushed ahead or behind in the queue under P). Then, Definition 4 follows from (4.1) by taking expectation under the assumption that c is the same across all busy periods and $\alpha = Var[W]^{FCFS}$. Note that if $\alpha = 0$ is chosen order fairness simply calculates the deviation $Var[W]^P - Var[W]^{FCFS}$, which is considered prominently in [3].

The above argument provides a strong justification for Definition 4 in the GI/D/1 setting. Further, using $Var[W]$ as a notion of fairness in the GI/D/1 case is quite intuitive since when job sizes are all the same it is 'fair' for all jobs to have the same waiting time, and $Var[W]$ captures the degree of variation of waiting times across jobs.

However, when moving beyond the GI/D/1 non-preemptive setting the justifications of $Var[W]$ must change because $Var[W]$ no longer satisfies the four axioms above. In fact, it turns out that the four axioms above are not simultaneously achievable. Additionally, the axioms above are only appropriate for use under non-preemptive scheduling policies (where the policy can be viewed as an

order on the jobs in the queue). The restriction to non-preemptive scheduling cannot be avoided, but it is possible to provide justification for why Definition 4 is still appropriate outside of the GI/D/1 setting.

In particular, though Definition 4 does not satisfy the four axioms outside of the GI/D/1 setting, the authors of [3] extend the applicability of order fairness by injecting a notion of proportional fairness into the measure. In particular, intuitively it is unfair to force a small job to wait behind a large job unless the large job has been in the system a long time before the small job arrives. Thus, the fairness of customers i and j with arrival times a_i and a_j should be dependent on the difference in the waiting times of these jobs, w_i and w_j. In particular, it should be some function $h(\cdot)$ that is increasing in $|w_j - w_i|$. Noting that

$$|w_j - w_i| = \begin{cases} |a_i - a_j - s_j|, & j \text{ is served ahead of } i \\ |a_i - a_j + s_i|, & \text{else} \end{cases}$$

it follows that running job j with service demand s_j ahead of job i is 'more fair' than running i ahead of j when $a_i - a_j > (s_j - s_i)/2$. Note that this adds a sense of proportional fairness to the measure since $|w_j - w_i|$ depends on the relative sizes of i and j. Using pairwise comparisons and taking $h(x) = c/2x^2$ calculations then show that this is equivalent to Definition 4.

As noted by the authors of [3], this generalization of order fairness to the GI/GI/1 queue is less than ideal because many of the original axioms no longer hold. For instance, when interchanging two jobs, the pairwise comparison of other jobs in the system is affected (i.e. the fourth axiom is violated). However, the simplicity of the resulting measure ($Var[W]^P$) is appealing since it allows for easy analysis.

In particular, the simplicity of the order fairness measure allows the easy comparison of many non-preemptive policies. For example, it is well known that FCFS minimizes $Var[W]$ among non-preemptive policies that do not use job size information [11], which highlights that FCFS is the most fair of such policies. So, it follows, for example, that in the GI/D/1 case FCFS is the most fair policy (as expected for temporal fairness measures). Further, in the GI/GI/1 setting it is not hard to see that FCFS is more fair than LCFS. Moving to size based policies, it is easy to see that SJF is more fair than LJF under this measure. However, surprisingly, when comparing SJF and FCFS, which is more fair depends on the job size distribution: If the job size distribution is nearly deterministic than FCFS is more fair than SJF, but if the job size distribution is highly variable then SJF is more fair than FCFS. This highlights that outside of the GI/D/1 order fairness is not purely a measure of temporal fairness.

4.2. Politeness

The second notion of temporal fairness that we discuss is 'politeness', which was introduced by Wierman in [52] and is defined as follows:

Definition 5. *Consider a GI/GI/1 queue. The* ***politeness experienced by a job of size*** *x under policy P, $Pol(x)^P$, is the fraction of the response time of a job of size x during which the seniority of the job is respected. A scheduling*

policy is ***impolite*** *under X and ρ if $\inf_x E[Pol(x)]^P = 1-\rho$, otherwise we say that P is* ***polite****.*

Note that, unlike most other measures in this paper, larger politeness is more fair. Informally, the idea behind Definition 5 is that a job is treated "politely" if it is unlikely that a later arriving job will bypass it in the queue (thus violating its seniority). In contrast to Order Fairness, politeness relies purely on this intuitive justification. However, this intuition certainly aligns with the idea of temporal fairness.

To begin to understand the definition, and in particular why the criterion of $1-\rho$ is used, it is useful to consider a few simple policies. Clearly, the politeness of FCFS is $Pol(x)^{FCFS} = 1$, which is the 'most polite' a policy can be. In contrast, PLCFS is the 'most impolite' policy and thus provides a lower bound on $E[Pol(x)]^P$ for all P. It can easily be proven that $E[Pol(x)]^{PLCFS} = 1-\rho$ [52]; thus we have that for all policies P, $E[Pol(x)]^P \in [1-\rho, 1]$. This is why $1-\rho$ is used as a criterion in Definition 5 to distinguish policies that are maximally impolite to some job size.

Now let us briefly discuss the politeness of some other common policies. Though the politeness of FCFS and PLCFS are independent of x, this is typically not the case. For example, $E[Pol(x)]^{PSJF} = 1-\rho(x)$ [52]. Thus, Definition 5 provides insight into the relative temporal fairness of different job sizes, which was not captured in the notion of Order Fairness.

One key benefit of politeness is that the analysis of scheduling policies can typically be performed quite easily. In particular, $E[Pol(x)]$ has been derived in the M/GI/1 setting for nearly all common scheduling policies, see [52]. To provide a peek into these results we mention three examples here. In an M/GI/1 queue with $E[X^2] < \infty$,

$$\begin{aligned} E[Pol(x)]^{PS} &= 1-\rho+\frac{\lambda}{x}\int_0^x t\overline{F}(t)dt \\ E[Pol(x)]^{FB} &= 1-\lambda\int_0^x \overline{F}(t)dt \\ E[Pol(x)]^{SRPT} &= 1-C_x^{SRPT}\rho(x), \end{aligned}$$

where $C_x^{SRPT} \in [0,1]$ is defined as

$$C_x^{SRPT} = \frac{\int_0^x \frac{\rho(t)}{\rho(x)}\frac{1}{1-\rho(t)}dt + \frac{\lambda\int_0^x t\overline{F}(t)dt}{(1-\rho(x))^2}}{\int_0^x \frac{1}{1-\rho(t)}dt + \frac{\lambda\int_0^x t\overline{F}(t)dt}{(1-\rho(x))^2}}.$$

These results highlight that often policies are polite to most job sizes; however, there is typically some particular job size that is treated as impolitely as possible, i.e. $E[Pol(x)] = 1-\rho$. For instance, PSJF, FB, and PS all have $E[Pol(x)] \to 1-\rho$ as $x \to \infty$, and are thus all impolite. Let us define the notions of Always Polite, Sometimes Polite, and Always Impolite, similarly to the case of Definition 2. Then, we can summarize the known results for common policies in Figure 4.

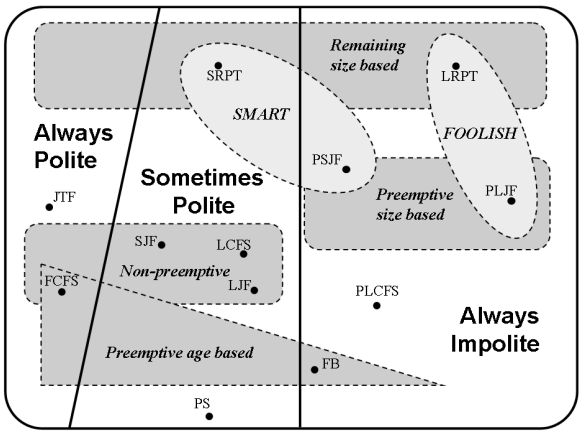


Figure 4: An illustration of the classification of common scheduling policies with respect to Definition 5 of temporal fairness.

A first observation about Figure 4, is that there are few common policies that are Always Polite. The one Always Polite policy other than FCFS is an uncommon policy called Jump To Front (JTF), which combines FCFS and PLCFS by allowing each arrival to jump to the front of a FCFS queue with probability p. See [52] for a further discussion.

A second important observation about Figure 4 is that it highligts a conflict between politeness and proportional fairness (as measured by Definition 1). In fact, the notion of politeness allows the formalization of this conflict in the following theorem, proven in [52]:

Theorem 5. *Consider an M/GI/1 queue with load ρ and a job size distribution having infinite support. If P is fair (according to Definition 1) then P is impolite and, specifically, $E[Pol(x)] \to 1 - \rho$ as $x \to \infty$.*

This result shows that it is impossible for a policy to be both temporally fair (as measured by Definition 5) and proportionally fair (as measured by Definition 1) for a given workload. Thus, there are no policies that are both Always Fair and Always Polite.

5. Combined measures of proportional and temporal fairness

Until this point we have presented fairness metrics that study either proportional fairness or temporal fairness in isolation and we have seen that these notions are often in conflict with each other. Now, we move to the discussion of measures that seek to provide one metric combining aspects of both proportional fairness and temporal fairness. We will survey results for two examples of such metrics that have received significant attention in the literature, the Resource

Allocation Queueing Fairness Measure (RAQFM) [40] and the Discrimination Frequency (DF) measure [45]. The interested reader can find other examples of fairness measures that combine proportional and temporal fairness studied in both the single and multi-server environments in [2, 44, 43].

It should be noted that measures combining proportional and temporal fairness can be categorized into two types: implicit combinations and explicit combinations. In an *explicit* combined metric, two individual measures focusing on either proportional fairness and temporal fairness in isolation are combined (typically in a weighted sum) to provide a combined fairness metric. Any of the measures we have discussed to this point could thus be combined to develop a combined measure. In contrast, an *implicit* combined metric inherently considers proportional fairness and temporal fairness together. Of the two measures we discuss here, RAQFM is an implicit combination and DF is an explicit combination. Each approach has advantages and disadvantages. For example, explicit combinations allow the relative weight of proportional fairness and temporal fairness to be shifted depending on the setting considered, while implicit combinations eliminate the need for an 'ad hoc' mixture of two unrelated metrics but enforce one specific combined view.

5.1. The Resource Allocation Queueing Fairness Measure (RAQFM)

The first combined fairness metric we discuss is an implicit combination called the Resource Allocation Queueing Fairness Measure (RAQFM) that was introduced by Raz, Levy, & Avi-Itzhak [40] and is defined as follows:

Definition 6. *Consider a GI/GI/1 queue. The* ***discrimination*** *of a job i, D_i is defined as $D_i = \int_{a_i}^{d_i} s_i(t) - 1/N(t)dt$ where a_i and d_i are the arrival and departure times of job i, $s_i(t)$ is the proportion of the service capacity given to job i and $N(t)$ is the number of jobs in the system at time t. The* ***unfairness of policy*** *P is defined as $Var[D]^P$, where D is the stationary discrimination of a job.*

The basic philosophy behind this measure is that at all times, every job in the system is worthy of an even share of the server capacity, i.e. at every time t, every job in the system deserves a service rate of $1/N(t)$. Notice that, as in Definition 1, this corresponds to viewing PS as the ideal for fairness. Given this notion of a 'fair share', the fairness of a policy is determined by looking at the variation from this fair service rate jobs experience. Note that it is the variation that is of interest because $E[D] = 0$ regardless of the work conserving policy being studied.[3]

In addition to the intuitive justification that each job's fair share of the server is $1/N(t)$ at time t, [40] provides a number of other justifications for Definition 6. In particular, [40] highlights that Definition 6 captures aspects of

[3]To see that $E[D] = 0$, note that whenever the system is busy at time t, $\sum_i (s_i(t) - 1/N(t)) = 0$ because $\sum_i s_i(t) = 1$.

both proportional fairness and temporal fairness through what they term the 'seniority preference test' and the 'service-requirement preference test'.

- *Service-requirement preference test*: If all jobs in the system have the same arrival time, then when serving jobs i and j with sizes s_i and s_j having $s_i < s_j$ it is more fair to complete job i before job j.
- *Seniority preference test*: If all jobs in the system have the same service times, then for jobs i and j with arrival times a_i and a_j having $a_i < a_j$ it is more fair to complete job i before job j.

In [40] it is shown that Definition 6 satisfies both the seniority preference test and the service-requirement preference test for the class of work-conserving, non-preemptive policies. Further, in the non-preemptive setting, it is straightforward to see that the impact of switching the order jobs i and j are serviced has on $Var[D]$ is monotonically increasing in both (i) the difference in the arrival times of i and j and (ii) the difference in the service demands of i and j. For more information on these and related properties, refer to the recent survey [4], which further discusses the psychological and philosophical motivations for Definition 6.

It is clear that RAQFM maintains many desirable properties. However, one drawback to RAQFM is that, as one can guess from the form of Definition 6, deriving $Var[D]$ is often a difficult task. In fact, closed form analyses of $Var[D]$ under most scheduling policies has proven elusive to this point, and represents an interesting open problem.

However, even without closed form analysis, it is possible to compare the fairness of policies using numerical techniques and via simulation, and a number of policies have been studied using these approaches, e.g., [4, 40]. Here, we survey some of the numerical results that have been attained in the M/M/1 setting.

In particular, let us first consider FCFS, LCFS, ROS, and PLCFS. In [40], it has been shown that in the M/M/1 setting

$$Var[D]^{PLCFS} \geq Var[D]^{LCFS} \geq Var[D]^{ROS} \geq Var[D]^{FCFS} \geq Var[D]^{PS} = 0$$

Further, it seems that $Var[D]^P \to 0$ as $\rho \to 0$ and that $Var[D]^P$ increases monotonically with ρ under all P.

One particularly interesting result that has been attained using RAQFM is the following. In a GI/GI/1 setting, [41] proves that under all policies that do not use size-based prioritization small jobs are discriminated against more than large jobs, i.e., the expected discrimination for jobs of size x is monotonic in x. This result can be viewed as a statement that in order for a policy to be fair it *must* give priority to small jobs. In addition, it is again possible to numerically analyze priority-based scheduling policies in the case of phase-type job size distributions, see [41] for the details. These results also highlight the fairness benefits that come from giving priority to small jobs.

One final word about RAQFM is that, though one drawback of the measure is that the analysis of scheduling policies is difficult, an important feature of

the measure is that it is quite easy to generalize beyond the single server environment. In fact, there are a number of interesting studies using extensions of RAQFM in multi-server and multi-queue systems, e.g., [39].

5.2. Discrimination Frequency

The final combined metric that we discuss is Discrimination Frequency, which was introduced by Sandmann in [45]. Unlike RAQFM, which implicitly captured the notions of proportional fairness and temporal fairness, Discrimination Frequency is defined as an explicit combination of two measures, one for proportional fairness and one for temporal fairness.

Definition 7. *Let n_i be the number of jobs that arrived no earlier than and completed no later than job i. Let m_i be the number of jobs with remaining size no smaller than job i when job i arrives that complete no later than job i. The Discrimination Frequency of job i is*

$$DF(i) = n_i + m_i.$$

In an M/GI/1 queue, ***unfairness of policy*** *P is defined as $E[DF]^P$.*

Notice that this definition takes a different approach than those of the other definitions we have discussed and simply counts the number of fairness violations. This is similar to the idea used in Gordon's earlier work measuring the 'skips and slips' of scheduling policies [19], which measured the frequency of overtaking (skips) and being overtaken (slips) in parallel server queues.

The first observation about this definition is that, intuitively, it is clear that n_i and m_i provide measures of temporal and proportional fairness respectively. Thus, one can easily adjust definition to balance the importance of proportional and temporal fairness as desired for any particular application. This flexibility is a key benefit of Discrimination Frequency.

A second observation is that, like RAQFM, Definition 7 satisfies the seniority-preference test and the service-requirement preference test.

A third important property of Discrimination Frequency is that, unlike RAQFM, the analysis of $E[DF]^P$ is typically simple enough to attain closed form results for most common scheduling policies. For example, it has been proven in [46] that

$$\begin{aligned}
E[DF]^{FCFS} &= \frac{\lambda^2 E[X^2]}{4(1-\rho)} + \rho - \lambda E[\min(X_1, X_2)] \\
E[DF]^{LCFS} &= \frac{\lambda^2 3E[X^2]}{4(1-\rho)} + \rho - \lambda E[\min(X_1, X_2)] \\
E[DF]^{SJF} &= \frac{\lambda^2 E[X^2]}{2} \int_0^\infty \frac{F(x)}{1-\rho(x)} dF(x) + \rho - \lambda E[\min(X_1, X_2)]
\end{aligned}$$

where X_1 and X_2 are i.i.d. service demands. Thus, we can see that $E[DF]^{SJF} \leq E[DF]^{FCFS} \leq E[DF]^{LCFS}$ for the M/M/1 setting. In addition to these results, the interested reader can find a variety of results (both simulation and analytic) for other policies in [45, 46].

6. Concluding remarks

The goal of this article is to provide a short survey of recent developments towards providing an analytic framework for studying the fairness of scheduling policies in a single server queue. Though the focus on fairness in the scheduling community is relatively recent, this survey highlights that there is a large and growing literature, which includes a number of surprising results. Further, the new understanding of fairness provided by the results surveyed here is already starting to have an impact on scheduling in practice. For example, in the setting of computer systems, there is a growing acceptance of the fact that giving priority to small jobs can actually be fair, which is leading to a wide array of new designs for applications such as web servers [20, 38], operating systems [14], routers [36, 37], wireless networks [22], transport protocols [56], and beyond.

However, it should be clear from this survey that many interesting and important questions remain. For example, in the setting of proportional fairness, it would be interesting to better understand the performance of FSP, which has withstood exact analysis for a number of years. Further, to this point the behavior of policies with respect to Definition 3 for proportional fairness in distribution is still poorly understood. Here a challenging open problem is to determine whether Definition 3 provides a min-max notion of fairness. In the context of combined fairness metrics, there are an abundance of open questions that remain. For example, deriving closed-form analyses of policies with respect to RAQFM and understanding the fairness of important common policies such as SRPT and PSJF under either RAQFM or DF. Finally, in the context of all of the fairness metrics we have discussed, the most important open questions still remain – is there a policy that can be near optimal for 'performance' (e.g. mean response time) while still being 'fair'? If so, what is the policy? If not, what is the fundamental limit? So far, these questions have not been answered in the context of any of the metrics discussed in this article.

Before ending, it is important to recall that the context of this article is quite limited – we have discussed fairness across jobs in the single resource context (i.e., a single-server queue). Our hope is that this limited focus allows the reader to see the complexities of defining and analyzing fairness in a simple setting, and that the ideas conveyed are useful in more complex environments. But, the reader should definitely be aware that there is also a large literature studying fairness outside of the single server queue in, for example, fairness across flows in multi-server systems and general queueing networks [24, 12, 27, 32, 7, 39, 44]. In each of these contexts there is, again, a large literature studying a variety of fairness metrics that have emerged in the last decade. Some of this work builds on the measures discussed in this survey, e.g., [39, 44]; however these more complex settings provide a variety of issues not present in the single server queue that motivate the use of different fairness metrics as well. For example, in the multi-server environment it may be important to be fair both to the customers and the servers, e.g., in a call center environment the dispatching scheme should be fair to the people answering the calls [1].

References

[1] M. Armony and A. R. Ward. Blind fair routing in large-scale service systems. Under submission.

[2] B. Avi-Itzhak, E. Brosh, and H. Levy. SQF: A slowdown queueing fairness measure. *Performance Evaluation*, 64:1121–1136.

[3] B. Avi-Itzhak and H. Levy. On measuring fairness in queues. *Adv. of Applied Probability*, 36(3):919–936, 2004.

[4] B. Avi-Itzhak, H. Levy, and D. Raz. A resource allocation fairness measure: properties and bounds. *Queueing Systems Theory and Applications*, 56(2):65–71, 2007.

[5] N. Bansal and M. Harchol-Balter. Analysis of SRPT scheduling: Investigating unfairness. In *Proc. of ACM Sigmetrics*, 2001.

[6] M. Bender, S. Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continous job streams. In *Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[7] T. Bonald and L. Massoulie. Impact of fairness on internet performance. In *Proc. of ACM Sigmetrics*, 2001.

[8] S. Borst, O. Boxma, R. Nunez-Queija, and B. Zwart. The impact of the service discipline on delay asymptotics. *Performance Evaluation*, 54:175–206, 2003.

[9] S. Borst, R. Nunez-Queija, and B. Zwart. Sojourn time asymptotics in processor-sharing queues. *Queueing Systems Theory and Applications*, 53(1-2), 2006.

[10] P. Brown. Comparing FB and PS scheduling policies. *Performance Evaluation Review*, 34(3):18–20, 2006.

[11] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.

[12] M. Dianati, M. Shen, and S. Naik. A new fairness index for radio resource allocation in wireless networks. In *Proc. of WCNC*, 2005.

[13] S. Doroudi and A. Wierman. Understanding fairness via a queueing game. Under submission.

[14] H. Feng, V. Misra, and D. Rubenstein. PBS: a unified priority-based CPU scheduler. In *Proc. of ACM Sigmetrics*, 2007.

[15] E. Friedman and S. Henderson. Fairness and efficiency in web server protocols. In *Proc. of ACM Sigmetrics*, 2003.

[16] M. Gong and C. Williamson. Quantifying the properties of SRPT scheduling. In *IEEE/ACM Symposium on Mod., Anal., and Sim. of Comp. and Telecomm. Sys. (MASCOTS)*, 2003.

[17] M. Gong and C. Williamson. Simulation evaluation of hybrid SRPT scheduling policies. In *Proc. of IEEE MASCOTS*, 2004.

[18] M. Gong and C. Williamson. Revisiting unfairness in web server scheduling. *Computer Networks*, 50(13):2183–2203, 2006.

[19] E. S. Gordon. *New problems in queues: Social injustic and server production management*. PhD thesis, Massachusetts Institute of Technology, 1987.

[20] M. Harchol-Balter, B. Schroeder, M. Agrawal, and N. Bansal. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems*, 21(2), 2003.

[21] M. Harchol-Balter, K. Sigman, and A. Wierman. Asymptotic convergence of scheduling policies with respect to slowdown. *Performance Evaluation*, 49(1-4):241–256, 2002.

[22] M. Hu, J. Zhang, and J. Sadowsky. A size-aided opportunistic scheduling scheme in wireless networks. In *Proc. of Globecom*, 2003.

[23] F. Kelly. *Reversibility and Stochastic Networks*. John Wiley & Sons, 1979.

[24] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness, and stability. *J. of the Operations Research Society*, 49, 1998.

[25] J. F. C. Kingman. The effect of queue discipline on waiting time variance. *Proc. of the Cambridge Philosophical Society*, 1962.

[26] L. Kleinrock. *Queueing Systems*, volume II. Computer Applications. John Wiley & Sons, 1976.

[27] C. E. Koksal, H. I. Kassab, and H. Balakrishnan. An anaysis of shortterm fairness in wireless media access protocols. In *Proc. of ACM Sigmetrics*, 2000.

[28] T. Lan, D. Gao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness. In *Proc. of Infocom*, 2010.

[29] R. Larson. Perspectives on queues: social justice and the psychology of queueing. *Operations Research*, 35:895–905, 1987.

[30] H. Levy, D. Raz, and B. Avi-Itzhak. Locality of reference and the use of sojourn time variance for measuring queue unfairness. *Operations Research Letters*, 35(3):311–318, 2007.

[31] M. A. Marson and M. Gerla. Fairness in local computing networks. In *Proc. of IEEE ICC*, 1982.

[32] J. Mo and J.Walrand. Fair end-to-end window-based congestion control. *Transactions on Networking*, 8(5):556–567, 2000.

[33] M. Nuyens and A. Wierman. The foreground-background queue: A survey. *Performance evaluation*, 65(3-4):286–307, 2008.

[34] M. Nuyens, A. Wierman, and B. Zwart. Preventing large sojourn times using SMART scheduling. *Operations Research*, 56(1):88–101, 2008.

[35] A. Rafaeli, G. Barron, and K. Haber. The effects of queue structure on attidues. *J. of Service Research*, 5(2):125–139, 2002.

[36] I. A. Rai, G. Urvoy-Keller, and E. Biersack. Analysis of FB scheduling for job size distributions with high variance. In *Proc. of ACM Sigmetrics*, 2003.

[37] I. A. Rai, G. Urvoy-Keller, M. Vernon, and E. W. Biersack. Performance modeling of FB based scheduling in packet switched networks. In *Proc. of ACM Sigmetrics-Performance*, 2004.

[38] M. Rawat and A. Kshemkalyani. SWIFT: Scheduling in web servers for fast response time. In *Symp. on Net. Comp. and App.*, 2003.

[39] D. Raz, B. Avi-Itzhak, and H. Levy. Fairness considerations in multi-server and multi-queue systems. In *Proc. of Valuetools*, 2006.

[40] D. Raz, H. Levy, and B. Avi-Itzhak. A resource-allocation queueing fairness measure. In *Proc. of ACM Sigmetrics-Performance*, 2004.

[41] D. Raz, H. Levy, and B. Avi-Itzhak. Class prioritization and server dedication in queueing systems: Discrimination and fairness. *Performance Evaluation*, 67:235–247, 2010.

[42] R. Rhodes, M. P. Battin, and A. Silvers. *Medicine and social justice*. Oxford University Press, 2002.

[43] G. Sabin, V. Sahasrabudhe, and P. Sadayappan. On fairness in distributed job scheduling across multiple sites. In *Proc. of Cluster*, 2004.

[44] G. M. Sabin. *Unfairness in parallel job scheduling*. PhD thesis, Ohio State University, 2006.

[45] W. Sandmann. A discrimination frequency based queueing fairness measure with regard to job seniority and service requirement. In *Proc. of Euro NGI Conf. on Next Generation Int. Nets*, 2005.

[46] W. Sandmann. Analysis of a queueing fairness measure. In *GI/ITG Conf. on Measurement, Modeling, and Eval. of Comp. and Comm. Sys.*, 2006.

[47] L. E. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16:678–690, 1968.

[48] A. Silberschatz and P. Galvin. *Operating System Concepts, 5th Edition.* John Wiley & Sons, 1998.

[49] W. Stallings. *Operating Systems, 2nd Edition.* Prentice Hall, 1995.

[50] A. Tanenbaum. *Modern Operating Systems.* Prentice Hall, 1992.

[51] A. Wierman. Fairness and classifications. *Performance Evaluation Review*, 34(4):4–12, 2007.

[52] A. Wierman. *Scheduling for today's computer systems: Bridging theory and practice.* PhD thesis, Carnegie Mellon University, 2007.

[53] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to unfairness in an M/GI/1. In *Proc. of ACM Sigmetrics*, 2003.

[54] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to higher moments of response time. In *Proc. of ACM Sigmetrics*, 2005.

[55] A. Wierman, M. Harchol-Balter, and T. Osogami. Nearly insensitive bounds on SMART scheduling. In *Proc. of ACM Sigmetrics*, 2005.

[56] S. Yang and G. de Veciana. Enhancing both network and user performance for networks supporting best effort traffic. *Transactions on Networking*, 12(2):349–360, 2004.