R U T C O R

RESEARCH

R E P O R T

QUANTIFYING FAIRNESS IN QUEUEING SYSTEMS: PRINCIPLES AND APPLICATIONS

Benjamin Avi-Itzhak[a] Hanoch Levy[b]
David Raz[c]

RRR 26-2004, **JULY 2004**

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/~rrr

[a] RUTCOR, Rutgers, the State University of New Jersey,
640 Bartholomew Road, Piscataway, NJ 08854-8003, USA,
aviitzha@rutcor.rutgers.edu
[b] School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel,
hanoch@cs.tau.ac.il
[c] School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel,
davidraz@post.tau.ac.il

RUTCOR RESEARCH REPORT

RRR 26-2004, JULY 2004

QUANTIFYING FAIRNESS IN QUEUEING SYSTEMS: PRINCIPLES AND APPLICATIONS

Benjamin Avi-Itzhak Hanoch Levy David Raz

Abstract. A fundamental and important property of a queue service discipline is its fairness. Recent empirical studies show fairness in queues to be highly important to queueing customers in practical scenarios. Despite this importance only little has been studied on this subject. The objective of this work is to illuminate the queue fairness issue and its dilemmas, and to overview the research conducted on this subject. We discuss the fundamental principles related to queue fairness in the perspective of the relevant real-life applications. The discussion is conducted in the context of a number of recently proposed queue fairness measures. We describe, discuss and compare their properties, and evaluate their relevance to the various practical applications.

1 Introduction

Why are we using ordered queues? Why do they serve in many real life applications, such as banks, supermarkets, airports, computer systems, communications systems, Web services, call centers and numerous other systems?

Perhaps the major reason for using a queue at all is to provide *fair service* to the customers. Furthermore, experimental psychology studies show that fair scheduling in queueing systems is indeed highly important to humans. Nonetheless, *Queueing Theory,* the theory that deals with analyzing queues and their efficient operation, has hardly dealt with the questions of what is a fair queue and *how fair* is a queueing policy.

The fairness factor associated to waiting in queues has been recognized in many works and applications. Larson (1988) in his discussion paper on the disutility of waiting, recognizes the central role played by 'Social Justice', (which is another name for fairness), and its perception by customers. This is also addressed in Rothkopf and Rech (1987) in their paper discussing perceptions in queues. Aspects of fairness in queues were discussed earlier by quite a number of authors: Palm (1953) deals with judging the annoyance caused by congestion, Mann (1969) discusses the queue as a social system and Whitt (1984) addresses overtaking in queues, to mention just three.

Empirical evidence of the importance of fairness of queues was recently provided in Rafaeli et. al. (2002),(2003) who studied, using an experimental psychology approach, the reaction of humans to waiting in queues and to various queueing and scheduling policies. The studies revealed that for humans waiting in queues the issue of fairness is highly important, perhaps some times more important than the duration of the wait.

This paper's aim is to present the subject of queue fairness, discuss the issues and dilemmas related to it, present the fundamental underlying assumptions and tie them to the real-life applications. When dealing with the introduction of a new measure for a quantity that is somewhat abstract and not very tangible, several questions should be brought up and discussed. What is the *physical entity,* or *performance objective* that should be dealt with? At what *level of detail* should the system be measured? What are the *physical properties* that affect the measure? How *intuitive* and *appealing* is the measure? And, how does the *measure relate* to the *relevant applications?* These questions are discussed and examined in this article in the context of a few fairness measures proposed recently in the literature.

What would be a fair service order in a supermarket queue or in an airport waiting line? Most people would instinctively respond that First-Come-First-Served (FCFS) is the fairest order, that is, serving jobs in increasing order of *seniority* is most fair. In fact, Kingman (1962) pronounces this in viewing FIFO (First-In-First-Out) as the 'fairest' queue discipline. This brings up the first factor playing a role in queue scheduling fairness, namely, *queue seniority.*

Aiming at understanding the problem better, we may pose the following "toy scenario", which some readers may associate with their own personal experience.
Mr. Short arrives at the supermarket counter holding only one item. Waiting at the queue he finds ahead of him Mrs. Long carrying a fully loaded cart of items. Would it be fair to have Mr. Long served ahead of Short and Short waiting for the full processing of Mrs. Long's loaded cart? Or, would it be more fair to advance Short in the queue and serve him ahead of Long?

This dilemma may cause some to "relax" their strong belief in the absolute fairness of FCFS. In fact, the dilemma brings to the discussion a new physical factor, that of *service requirement.* The basic intuition thus suggests that prioritizing short jobs over long jobs may also be fair. It is the trade-off between these two physical factors, *seniority* (prioritize Mrs. Long) and *service requirement* (prioritize Mr. Short), that creates the dilemma in this case. This tradeoff, as well as the "Long vs. Short" scenario, will accompany us in this paper in attempting to understand fairness in queues.

What bothers one in a queue? What is the *performance objective* one aims for while staying in a queue? Understanding this issue should form the basis for proposing a proper fairness measure, since the measure must be built around the *performance objective* of interest to the queue customers. This question is discussed in Sections 3 and 4, following the presentation of the model in Section 2. First, in Section 3, we note that this article focuses on *job-driven* systems. We describe what job driven systems (as opposed to flow based systems) are, and review the real-life applications that are associated with these models.

Second, in Section 4, we discuss the performance objectives associated with job-driven systems. We claim that in addition to the natural candidate, namely that of *waiting times (delays),* one may have to consider, in some cases, the performance objective of *service* (that is whether a service is granted to the job or not) which might be important alongside the waiting times. It should be noted that queueing theory has been mostly occupied with the performance metrics of waiting time and dealt less with the metrics of service (see e.g. text books on the subject: Kleinrock (1975, 1976), Hall (1991), Cooper (1981), Daigle (1992), Gross and Harris (1974).

Having dealt with the performance objective, we then (in Section 5)) ask at what granularity level the performance metrics should be dealt with. Our conclusion is that it is desirable to deal with the performance metrics at three granularity levels, *individual discrimination, scenario fairness* and *system fairness*. The addressing of fairness at all three levels is similar to the addressing of the waiting time measure that can also be evaluated at these three levels. More importantly, this allows theoreticians and practitioners, as well as queue users, to develop good feel and intuition towards the measure, which will assist in getting used to the measure and to using it.

The physical properties which are at the heart of queue scheduling in general and fairness in particular, namely *seniority* and *service requirement* are posed and discussed in Section 6. These are best illuminated via the "Short vs. Long" example presented above. The tradeoff between serving Short first and serving Long first reflects the tradeoff between seniority and service

requirement. This translates into the modeling dilemma (and difficulty) of how to account for both seniority and service requirement in quantifying fairness, a dilemma that seems to be in the heart of fairness quantification models.

Having discussed what fairness measures should quantify, and the seniority and service requirement factors, we then (in Section 7) review the approaches for quantifying fairness in (job based) queueing systems. Analytic treatment and quantification of queue fairness have been quite limited in the literature, and been addressed only very recently. Three references that propose measures or a criterion for fairness of queues are Avi-Itzhak and Levy (2004), Raz, Levy and Avi-Itzhak (2004), who propose measures, and Wierman and Harchol-Balter (2003), who propose a criterion. The modeling dilemma of seniority versus service requirement seems to be at the heart of these queue fairness modeling attempts: The approach proposed in Avi-Itzhak and Levy (2004) centralizes on the *seniority* factor. In contrast, the approach proposed by Wierman and Harchol-Balter (2003), focuses on the *service-requirement* factor. Lastly, in attempting to give *seniority* and *service requirement* even treatment, the approach of Raz, Levy and Avi-Itzhak (2004) focuses on neither of them and chooses to focus on a third factor, that of *resource allocation.*

Lastly (in Section 8), after discussing the properties of the proposed fairness measures, we focus on the real-life applications and examine how (and whether) the various queueing fairness measures apply to the various applications. This issue, too, seems to strongly tie to the *seniority* versus *service-requirement* dilemma: The selection of a proper quantification approach to an application depends on how strong are the roles *seniority* and *service requirement* play in the application.

1.1 Additional Related Work

Much work has been conducted in the context of communications networks where the concern is with *flows* traversing a communications node and in allocating the *bandwidth fairly* among the *flows.* This is in contrast to the present work that focuses on fairness to *jobs*. One of the earliest attempts to define flow-fairness is Wang and Morris (1985) where the Q-factor is defined. Later on, the research on flow fairness has flourished with the introduction of *Weighted Fair Queueing* (WFQ), which deals with the fair scheduling of packet flows. Some early papers on the subject are Demers, Keshav and Shenker (1990), Greenberg and Madras (1992), Parekh (1992), Parekh and Gallager (1993), (1994), Golestani (1994), Rexford, Greenberg and Bonomi (1996), Bennet and Zhang (1996). Many other papers have been published on this subject. A popular measure of fairness within that context is the *relative fairness bound* (Used by Golestani (1994) and others) which captures the maximum possible difference between the (normalized) service received by any two streams. As such it measures “fairness of throughput of streams”.

2 Model

We consider a general queueing system consisting of a single server (in some cases we will consider multiple servers). Jobs, denoted $J_1, J_2, \ldots$ arrive at the system at arbitrary *arrival epochs*, denoted $a_1, a_2, \ldots$, respectively. In many queueing models, jobs are associated in a one-to-one manner with customers (to be denoted $C_1, C_2, \ldots$). For convenience of notation we assume that $a_i \le a_{i+1}$. Job J_i requests some service at the server, the amount of which we denote by s_i; For simplicity, we will measure the service requirement in units of time. The server grants service to the customers according to some scheduling policy. Once J_i receives its full amount of service s_i (which does not have to be given continuously or at full rate) it leaves the system, and the epoch when it leaves is called its *departure epoch*, denoted d_i. The duration J_i stays in the system is called *system time* and is denoted $t_i = d_i - a_i$. The duration J_i waits and does not get service is the *waiting time* of J_i, denoted by w_i and is given by $w_i = t_i - s_i = (d_i - a_i) - s_i$, except for processor sharing disciplines, where this conventional definition of *waiting time is not applicable*. These notations a_i, s_i, d_i, t_i, w_i are used to denote the actual values, attributed to J_i in a specific sample path of the system. The same letters capitalized are used to denote the corresponding random variables.

3 Job-Based systems and their Corresponding Applications

Queueing model applications can be classified into 1) *Job-based systems*, and 2) *Flow-based systems*. In the former, each customer, say C_i, is associated with a single job J_i. Of interest is therefore the performance experienced by that job. In the latter, customer C_i is associated with a stream (or flow) of jobs $J_i^1, J_i^2, \ldots$ arriving at epochs $a_i^1, a_i^2, \ldots$ respectively. Of interest is the performance experienced by the whole flow. The applications associated with this latter model are communications networks applications where a customer (sometimes called source) is associated with a stream of packets that are sent through a communications device, e.g., a router.

Our focus in this work is on job-based systems. Applications that are associated with this model are:

1. **Banks, supermarkets, public offices and the like**, in which customers physically enter a queue where they wait for their service and then get served. In the supermarket, for instance, the job is the processing of the *whole* customer's cart including the payment process. Once being served the customer leaves the system.
2. **Computer systems,** in which a customer (or a customer's computer application) submits a job to the system and the customer gets satisfied when the service of the job is completed.
3. **Call centers,** in which customers call into a call center to receive service, possibly wait in a virtual queue (while listening to some music) until being answered by "the next available agent". Call center queueing systems are conceptually identical to physical queueing

facilities, such as banks, except that the service is done over the phone, and the customer waits on the telephone line (instead of physically waiting in a physical line). Other differences are that the system's operator has much more control over the scheduling process, and that, unless special technology is applied, other customers are not seen by the customer.

4 What is the performance issue: Delay vs. Service

What performance measure should be accounted for when quantifying queue fairness? The immediate and most natural candidate is the *job delay,* which is either the waiting time or the waiting plus service time experienced by the job. This has been the main quantity (perhaps almost the sole quantity) used in queueing theory to evaluate queueing systems (see, e.g. text books on the subject, Kleinrock (1975, 1976), Hall (1991), Cooper (1981), Daigle (1992)) and is frequently being looked at via the expected delay or its variance in steady state. Under this quantity, customer satisfaction decreases with the delay experienced by the job and thus customer's objective is to minimize delay. The use of this quantity seems to be appropriate when the major performance issue associated with job queueing is indeed the delay experienced in the system. As an example, consider the waiting line in a supermarket where the annoyance of arriving late to a queue is merely due to the waiting in queue, since the service itself (purchasing the products) is guaranteed.

While delay has been the main performance metrics used by queueing theory, one should also consider *job service*, which received much less attention. By job service we refer to the actual service (*not* service time) given to the job by the system. To understand this performance objective, one should consider systems where the service is not guarantee, .e.g., systems where the service includes the selling of a finite-quantity product or systems where the server is shut down at a predetermined time. In these systems, customers whose processing by the server is delayed (e.g., due to the prioritization of other customers) may encounter the situation in which the product is not available, or the server is shut down before their turn to be served comes, and thus they experience service degradation. In a simplistic representation, this can be a zero-one variable where zero means that no service is given and one means that a full service is given. The use of this variable seems to be appropriate when the major performance issue is that of service, for example a queueing line for scarce concert tickets.

Note that both quantities, delay and service, are affected by scheduling decisions and thus can be the subject of a fairness measure.

5 At what Granularity Level Should Fairness be Quantified and Measured

In quantifying a physical property of a queueing system we identify three granularity classifications of the measure: The *job-individual measure,* the *scenario measure,* and the *system measure.* All three play a role in traditional queueing analysis, e.g., in dealing with system *delays*. In the context of delays, these are, respectively, the delay experienced by a specific job, the average delay when computed over a finite set of jobs under a specific scenario and the

expected delay in the system under steady state. We may define these measures, in the context of queue fairness as:

1. ***Job-Individual Discrimination:*** This is a quantity attributed to the individual job (customer). It represents the performance experienced by a specific job (customer) under a specific scenario (or a sample path). For example, consider the discrimination experienced either by Short or by Long in the Long vs. Short case.
2. ***Scenario (Sample path) fairness:*** A summary-statistics that summarizes the performance as experienced by a (finite) set of jobs under a particular scenario (a sample path). For example, consider some averaging of the discriminations experienced by Long and Short and the other jobs present at the system at that time.
3. ***System fairness***: A summary statistics of a probabilistic measure (e.g. expected value or variance) of the performance as experienced by an arbitrary job, when the system is in steady state. This can be extended to a similar measure under transient behavior of the system.

It should be noted that queueing theory has dealt explicitly mainly with the third type of quantity (expected delay or its variance), as the other quantities are somewhat trivial in the context of customer delay. In the context of fairness, it is nonetheless important to make explicit use of the job-individual and scenario quantities as well, since humans can feel them better and associate with them better than with the third quantity. This is important to building confidence in the fairness measure, which is somewhat abstract, non-tangible and difficult to feel.

6 The Basic Physical Entities behind Queue Fairness: Seniority and Service Requirement

What are the basic physical quantities playing a role in queue fairness? Two fundamental quantities determine the queueing process and the job scheduling decisions. These are the arrival epochs and service times, a_i, s_i. As our goal is to focus on the pure queueing process and neutralize other external parameters, we will deal with these variables only. To this end, we are not accounting for external parameters, such as payments made by customers or a gold/silver/bronze classification of customers.

Since these quantities are the only ones determining the queueing and scheduling process, they also serve as the fundamental variables for determining scheduling fairness. For convenience of presentation, we will translate the arrival time epoch and get the following two basic physical quantities: 1) *Seniority*, and 2) *Service requirement*. The seniority of J_i at epoch t is given by $t - a_i$. The service requirement of J_i is s_i. One may recall that *seniority* and *service-requirement* were in the heart of the dilemma in the Short vs. Long scenario.

It is natural to expect that a "fair" scheduling discipline will give preferential service to highly senior jobs, and to low service-requirement jobs. This can be stated formally in the following two fundamental principles:

1. *(Weak) Service-requirement Preference Principle*: If all jobs in the system have the same arrival time, then for jobs J_i and J_j, arriving at the same time and residing concurrently in the system, if $s_i < s_j$ then it will be more fair to complete service of J_i ahead of J_j than vice versa.
2. *(Weak) Seniority Preference Principle*: If all jobs in the system have the same service times, then for jobs J_i and J_j, residing concurrently in the system, if $a_i < a_j$ then it will be more fair to complete service of J_i ahead of j_j than vice versa.

A stronger form of the preference principles is as follows:

1. *Strong Service-requirement Preference Principle:* For jobs J_i and J_j, arriving at the same time and residing concurrently in the system, if $s_i < s_j$ then it will be more fair to complete service of J_i ahead of J_j than vice versa.
2. *Strong Seniority Preference Principle:* For jobs J_i and J_j, residing concurrently in the system and requiring equal service times, if $a_i < a_j$ then it will be more fair to complete service of J_i ahead of J_j than vice versa.

The seniority preference principle is rooted in the common belief that jobs arriving at the system earlier "deserve" to leave it earlier. The service-requirement preference principle is rooted in the belief that it is "less fair" to have short jobs wait for long ones.
It should be noted that when $a_i < a_j$ and $s_i > s_j$ (the Short vs. Long case) the two principles conflict with each other, and thus the relative fairness of the possible scheduling of J_i and J_j is likely to depend on the relative values of the parameters.

One may view these two preference principles as two axioms expressing one's basic belief in queue fairness. As such, one may expect that a fairness measure will follow these principles. A fairness measure is said to follow a preference principle if it associates higher fairness values with schedules that are more fair. A formal definition is given next:

Definition: Consider jobs J_i and J_j, requiring equal service times and obeying $a_i < a_j$. Let π be a scheduling policy where the service of J_i is completed before that of J_j and π' be identical to π, except for exchanging the service schedule of J_i and J_j. A fairness measure is said to adhere to the strong seniority preference principle if the fairness value it associates with π is higher than that it associates with π'.

Similar definitions can be given to the service-time preference principle and to the weak-versions of the preference principles.

It is easy to see that if a fairness measure adheres to the *strong preference principle* (either Service-requirement or Seniority) then it must adhere to the corresponding *weak preference principle*.

6.1 Scheduling Policies and the Preference Principles

To illustrate the preference principles in the context of scheduling policies we review several common policies and examine whether they follow the preference principles. A formal definition is:

Definition: A scheduling policy π is said to follow the strong seniority preference principle if for every two jobs J_i and J_j, requiring equal service times and obeying $a_i < a_j$, π completes the service of J_i ahead of that of J_j.

A similar definition can be given for the strong service-time preference principle and for the two weak preference principles.

Using these definitions, one can classify common scheduling policies as follows:

1. ***FCFS***: The *First-Come-First-Served* scheduling follows the strong seniority preference principle. On the other hand, since it gives no special consideration to shorter jobs, it does not follow the service-time preference principle (weak or strong).
2. ***LCFS*** and ***ROS***: The *Last-Come-First-Served* and *Random order of Service* policies do not follow the seniority preference principle (either strong or weak). Further, neither do they follow the service-time preference principle.
3. ***SJF*** and ***LJF:*** The *Shortest Job First (SJF)* follows the strong service-time preference principle. Nonetheless – it does not follow the seniority preference principle (both strong and weak). The *longest Job First (LJF)* follows none of the principles.
4. ***PS*****:** The *Processor Sharing* policy follows both the strong seniority preference principle and the strong service-time preference principle.
5. ***FQ:*** *Fair Queueing*, which is the non-weighted version of Weighted Fair Queueing (Parekh (1992) and Parekh and Gallager (1993)), serves the jobs in the order they complete service under Processor Sharing (unless some of the jobs are not present at the server at the time that the service decisions must be taken). This property and the fact that PS follows both of the strong preference principles, imply that FQ follows both the strong seniority preference principle and the strong service-time preference principle.

7 Fairness Measures: A Review of Proposed Measures and their Properties

Having discussed the performance issues associated with queue fairness, we next turn to review recent measures proposed in the literature. We will examine how these measures treat the basic performance issues and how they fit with the various applications.

7.1 Order Fairness

The order fairness measure was studied in Avi-Itzhak and Levy (2004). The basic underlying model used in that study assumes that all service times are identical. In that context the major factor of interest is that of job-seniority. The study deals with a specific sample path of the system, and examines a realization π of the service order (that is, a feasible sequence of job indices reflecting the order of service), and with a fairness measure $F(\pi)$ defined on the service order. The paper assumes several elementary axioms on the properties of $F(\pi)$. The major axiom is:

1. ***Monotonicity of F() under neighbor jobs interchange*:** If two neighboring jobs are interchanged to modify π and yield a new service order π' then $F()$ increases if the interchange yields advancing the more senior of the two jobs ahead of the less senior job, and it decreases if the interchange advances the less senior job ahead of the more senior job. If the seniority of the interchanged jobs is the same – $F()$ is not affected by the interchange.

The additional axioms deal with 2) *Reversibility of the interchange*, 3) *Independence on position and time,* and 4) *Fairness change is unaffected by jobs not interchanged.*

The reader may recognize that the core axiom of this approach, Axiom 1, is simply a mathematical form to express the *seniority preference principle* presented in Section 6.

The results derived in Avi-Itzhak and Levy (2004) show that for a specific sample path the quantity c $\sum_i a_i \Delta_i$ +α, where Δ_i is the *order displacement* of J_i (number of positions J_i is pushed ahead or backwards on the schedule), and c>0 and α are arbitrary constants satisfying the basic axioms. This quantity is the unique form satisfying the axioms applied to any feasible interchange (not necessarily of neighbors).Under steady state this quantity is equivalent to the *variance* of the *waiting time* (with a negative sign). Thus, when all *service times* are *identical* the *waiting time variance* can serve as a surrogate for the *system's unfairness measure*.

7.1.1 Properties

The main properties of this measure are:

1. The measure adheres to the strong *Seniority Preference Principle* (Section 6). This can be verified by recalling that the unfairness function for a sample path is given by $\sum_i a_i \Delta_i$ and by examining the change of this function due to the exchange of J_i and J_j.
2. When all service times are identical, the fairest policy in the family of work conserving and uninterrupted service policies is FCFS. The most unfair policy under these conditions is LCFS.
3. The measure does not adhere[1] to *Service-requirement Preference Principle* (Section 6): If one uses the variance of waiting time as a measure of unfairness, then there are cases where it is more fair to serve a long job ahead of a short job. For example consider a system with two jobs only, J_1 and J_2 whose service times are $s_1 = 1, s_2 = \varepsilon \to 0$. Serving the longer job J_1 first leads to a waiting time variance that approaches 0 while serving the shorter job J_2 first leads to a waiting time variance that approximately equals 1/4.

7.2 Normalized-delay based fairness

Normalized-delay based fairness was presented in Wierman and Harchol-Balter (2003) in which a fairness criterion (as opposed to a fairness measure) was proposed. The aim of this criterion is to address the differences in service times between different jobs and to follow a principle under which short jobs should be given some preferential service over long jobs (similar to the Service-time preference principle).
Under the criterion proposed in Wierman and Harchol-Balter (2003) each job is characterized by its service time only. The measure of interest is the "slow down" $S(x) \stackrel{def}{=} T(x)/x$ where x is the service time and $T(x)$ is a random variable denoting the delay (response time) experienced by a customer whose service time is x. The expected slowdown for a job of size x is $E[S(x)]$, and a scheduling policy is said to be fair for given load and service distribution if $E[S(x)] \leq 1/(1-\rho)$ for all values of x, where ρ is the system's load. A service policy is *always fair* if it is fair under all loads and all service distributions. A service policy is *always unfair* if it is not fair under all loads and all service distributions. Other policies are *sometimes unfair*.

7.2.1 Properties

The properties of this criterion are:

1. The criterion classifies a large class of service disciplines, common in computer systems, into "always fair" "always unfair" and "sometimes fair". A few examples are:
    a. **Always fair:** Processor Sharing (PS) and Preemptive LCFS.

[1] In fact, it might not be appropriate to examine this principle as the measure is built for equal service-time situations.

   b. **Always unfair:** All non-size based non-preemptive policies, in particular FCFS. Also age-based polices are always unfair, in particular Feedback Scheduling (FB).
   c. **Sometimes Unfair:** Shortest Remaining Processing Time (SRPT).
2. The criterion is relatively easy to apply for general service time distributions (M/G/1 type systems) as the measure of interest is *E[T(x)/x]*.
3. The *Seniority Preference Principle* does not hold. Specifically, the classification stated above implies that under this criterion FCFS is "always unfair" while LCFS preemptive is "always fair"; these predictions contradict the *Seniority Preference Principle*.
4. Intuitively speaking, this criterion may possibly adhere to the *Service time Preference Principle* if the criterion is extended to be a measure and after some adaptations. We make this intuitive statement based on recognizing that the criterion favors prioritization of short jobs over long jobs. However it is an open question whether this preference principle always holds and under what formulation.

7.3 Resource Allocation based Fairness

A Resource Allocation Queueing Fairness Measure (RAQFM) was introduced in Raz, Levy and Avi-Itzhak (2004). The measure aims at accounting both for seniority and service-requirements, and does it by focusing on the fair sharing of the system resources. The method can apply to multiple servers, but for the sake of presentation will be described for a single server system.
The basic philosophy behind the method is that at every epoch *t* at which there are *N(t)* jobs present in the system, they all are entitled to an equal share of the server's time. Thus, the temporal *warranted service rate* to be given to a job at that epoch, is given by: *1/N(t)*. The overall warranted service of job *i* is given by integrating this value over the duration that J_i stays in the system. Subtracting this warranted service from the granted service (which is the service granted to J_i, namely its service time, s_i) yields the *discrimination* of J_i, denoted $\delta_i = s_i - \int_{a_i}^{d_i} \frac{1}{N(t)} dt$. Note that the discrimination may be positive or negative. Taking summary statistics over all discriminations experienced by the customers yields an unfairness measure for the system. This measure can apply to a specific scenario (sample path), to yield the unfairness of that path. Similarly, taking expectations of this measure over all sample paths yields the system unfairness. One of the basic properties of the discrimination function is that it is a zero-sum function (namely the total discrimination in the system, at every epoch, is 0). Thus, the expected value of discrimination is meaningless, and the proper summary statistics is the second moment (or variance) or expected absolute value of discrimination.

7.3.1 Properties

The main properties of this measure are:

1. The measure adheres to the *Strong Seniority Preference principle* (Section 6), for work conserving and uninterrupted service policies. This is proven in Raz, Avi-Itzhak and Levy (2004a).
2. When all service times are identical, the fairest policy in the family of work-conserving and uninterrupted-service policies is FCFS. The most unfair policy under these conditions is LCFS.
3. The measure adheres to the weak *Service-requirement Preference Principle* (Section 6), for work conserving and uninterrupted service policies. Nonetheless, it does not adhere to the strong version of this principle, as there exist some counter examples. These properties are proven in Raz, Avi-Itzhak and Levy (2004a).
4. The measure yields to analysis for the family of Markovian (M/M type) queues. It is an open subject of research whether (and how) it yields to analysis for general service time (e.g. M/G/1) type systems.

7.4 Summary of Major properties

The two measures and the criterion can be roughly classified according to their treatment of the seniority and service time physical properties. This summary is depicted in Figure 1 where, in the first column, we indicate the major focus of the measures: Order fairness accounts mainly for seniority, normalized-delay fairness accounts mainly for the service times, and resource allocation accounts for both. The second column of the figure indicates what type of applications would fit to these measures: Order fairness fits service sensitive applications while normalized-delay fairness and resource-allocation fairness fit delay sensitive applications.

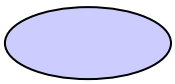
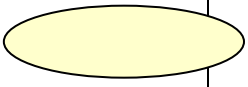
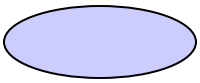
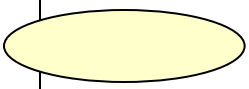
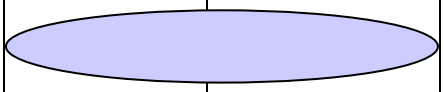
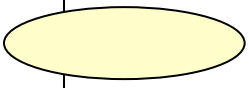
Measure/ Criterion	Physical quantities accounted for		Can treat Applications which are sensitive to	
	Seniority	Service time	Service delivery	Delay
Order fairness measure (7.1**)**				
Normalized delay criterion (7.2)				
Resource allocation measure (7.3**)**				

Figure 1: Summary of major properties of fairness measures

8 Application Perspective

How do the measures outlined above fit with the various queueing applications? This, as we discuss next, depends on the characteristics of the applications. The mapping of applications to fairness measures is given (partially) in Figure 2.

Some applications are characterized by high sensitivity to *service* and lower sensitivity to *delay;* these are outlined in the first row of applications in Figure 2. Perhaps most of the "historical queueing systems", e.g., the waiting line for bread, can be categorized under this category (this "historical" queueing experience is perhaps the reason for having many people strongly believing in the notion of servicing customers by order of arrival). Today's applications are waiting lines for limited-supply products, such as the queue for highly demanded concert-tickets. Another application is the waiting line in a call center specializing in selling airline tickets, in which often some of the tickets (e.g., special price or special date tickets) are at very low supply. The fairness of these applications is very sensitive to job seniority and is less sensitive to service times (3rd and 4th columns)
For these service sensitive (seniority sensitive) applications *order fairness* (Section 7.1) fits well, as it focuses on job seniority. Recall, however, that that measure assumes identical service times. Thus, its use in the case of non-identical service times, though sounds reasonable, still needs to be studied and understood.

Other applications are characterized by high sensitivity to delay and low sensitivity to service (as the service is more or less guaranteed). These, in fact, form the majority of today's applications.

Within this class we first distinguish applications where the service times of all customers are more or less identical (nearly deterministic); these are denoted on the second row of Figure 2. These include, for example, waiting lines for (unmarked) theater tickets. In this case, even if the supply is large and thus the performance is sensitive mainly to delay, one can apply the *order fairness* (Section 7.1), since service times are more or less identical. One can also apply in this case the resource allocation measure (Section 7.3) as it can handle this equally well.

Second, within this class of delay sensitive applications, we distinguish applications where the service time varies across customers (third line in Figure 2). These include waiting lines in supermarkets, airlines counters, public offices, and call centers with unlimited products. Here customers will be sensitive both to seniority and to service times. In these applications neither order-fairness (does not account for service time differences) nor normalized-delay fairness (does not account for seniority) can be used; the *resource allocation fairness measure* (Section 7.3), which accounts both for service times and seniority, is the most appropriate

Within the class of delay-sensitive applications one may recognize some applications where the customers are *not aware* of the relative seniority of the jobs in the system (fourth line of Figure 2). These may include jobs performed in a computer system where the customers who submit the jobs cannot know the relative seniority/status of their jobs. In such applications, the blindness of customers to relative seniority may allow one to use the *normalized-delay fairness* approach

(Section 7.2). Similarly, the "blindness" of customers to service requirements of other customers may allow the use of the *order fairness* measure.
Note however, that even under the blindness conditions, it is likely that customers will require the system not to be seniority-blind or service requirement-blind, namely not to use the normalized-delay fairness. That is, even if justice cannot be seen, customers may want it to be done.

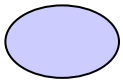
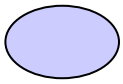
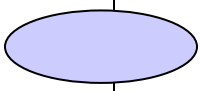
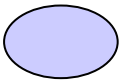
Application characteristics	**Applications**	**Fairness sensitive to**		**Performance Objectives**	**Applicable Fairness Measure**
		Seniority	**Service time**		
Service sensitive	Airline reservation Call center, "Line for bread"			Service	Order fairness (7.1)
Delay sensitive, identical service times	Airport immigration lines			Delay	Order fairness (7.1) + Resource allocation (7.3)
Delay sensitive, variable service time	Call centers, Supermarkets, Banks, Computer systems			Delay	Order fairness (7.3)
Delay sensitive, seniority-blind customers	Computer systems			Delay	Normalized-delay (7.2) + resource allocation (7.3)

Figure 2: Fairness-related Properties of Applications and the applicable measures

9 Concluding Remarks

We argued that fairness is a fundamental property of queueing systems and that it is highly important to customers. Little work has been done on this subject in the past; an increase in research in this area occurred in recent years, which contributed to better understanding of the subject. Nevertheless, more research must be conducted to have a good understanding of the

issue. For example, there exist a huge number of queueing systems and queueing scheduling policies, which were studied in the past and where the focus has been on the delay of the individual customer. Fairness evaluation of these systems will contribute greatly to the understanding of the relative benefits of these systems.

10 References

1. B. Avi-Itzhak and H. Levy (2004). On measuring fairness in queues. *Advances of Applied probability.* To appear, 2004.
2. E. G. Coffman, Jr., R. R. Muntz, and H. Trotter (1970), Waiting time distribution for processor-sharing systems. *JACM*, 17:123-130, 1970.
3. R. B. Cooper (1981), Introduction to Queueing Theory, Macmillan, 1972. Second Edition, North –Holland (Elsevier), 1981. Also at http://www.cse.fau.edu/~bob/publications/IntroToQueueingTheory_Cooper.pdf.
4. J. D. Daigle (1992), Queueing Theory for Telecommunications, Addison-Wesley, September, 1991. Second Printing, Spring 1992.
5. A. Demers, S. Keshav, and S. Shenker (1990), Analysis and simulation of a fair queueing algorithm. *Internetworking Research and Experience*, 1:3-26, 1990.
6. A. G. Greenberg and N. Madras (1992), How fair is fair queueing? *JACM*, 3(39):568-598, 1992.
7. D. Gross and C.L. Harris, Fundamentals of Queueing Theory, Wiley & Sons, New York, 1974.
8. R. W. Hall (1991), Queueing Methods for Services and Manufacturing, Prentice Hall, 1991.
9. M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal (2003), Size-based scheduling to improve web performance, *ACM Transactions on Computer Systems,* 21(2):207-233, May 2003.
10. L. Kleinrock (1975), Queueing Systems Vol I: Theory, ed. John Wiley & Sons, New York, 1975.
11. L. Kleinrock (1976), Queueing Systems Vol II: Computer Applications, Wiley-Interscience Publication.
12. J. F. C. Kingman (1962), The Effect of Queue Discipline on Waiting Time Variance, *Proc. Camb. Phil. Soc*., 58:163-164, 1962.
13. R. C. Larson (1987), Perspective on queues: Social justice and the psychology of queueing. *Operations Research.* 35:895-905, Nov-Dec 1987.
14. I. Mann (1969), Queue culture: The waiting line as a social system, *Am. J. Sociol*. 75:340-354, 1969.
15. A. Parekh (1992). A generalized processor sharing approach to flow control in integrated services networks. Ph.D Dissertation, MIT, 1992.
16. A. Parekh and R. G. Gallager (1993). A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. Networking*, 1:344-357, June 1993.

17. A. Parekh and R. G. Gallager (1994), A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. Networking*, 2:137-150, 1994.
18. A. Rafaeli, G. Barron, and K. Haber (2002), The effects of queue structure on attitudes. *Journal of Service Research*, 5(2):125-139, 2002.
19. A. Rafaeli, E. Kedmi, D. Vashdi, and G. Barron (2003). Queues and fairness: A multiple study investigation. Faculty of Industrial Engineering and Management, Technion. Haifa, Israel. Under review, 2003.
20. D. Raz, H. Levy and B. Avi-Itzhak (2004), A Resource-Allocation Queueing fairness Measure, In *Proceedings of ACM Sigmetrics/Performance conference, June 2004, New York*, pages 130-141. Also appears in *Performance Evaluation Review* Special Issue Volume 32 No. 1.
21. D. Raz, B. Avi-Itzhak, and H. Levy (2004a), Properties and Bounds of RAQFM. *In preparation.* 2004.
22. D. Raz, B. Avi-Itzhak, and H. Levy (2004b), Classes, priorities and fairness in queueing systems. Technical Report RRR-21-2004, RUTCOR, Rutgers University, June 2004. *http://rutcor.rutgers.edu/pub/rrr/reports2004/21_2004*
23. M. H. Rothkopf and P. Rech (1987), Perspectives on Queues: Combining Queues is not Always Beneficial. *Operations Research*, Vol 35, No. 6, November-December 1987.
24. M, Shreelhar and G. Varghese (1996), Efficient Fair Queueing Using deficit Round Robin**,** *IEEE/ACM Transactions on Networking*, Volume 4, Issue 3 (June 1996), pp.375 – 385.
25. Y.T. Wang and R.J.T Morris (1985), Load sharing in distributed systems, *IEEE Tx. on computers*, Vol. C-34, No. 3, pp. 204-217. 1985.
26. W. Whitt (1984). The amount of overtaking in a network of queues. *Networks,* 14(3):411-426, 1984.
27. A. Wierman and M. Harchol-Balter (2003), Classifying scheduling policies with respect to unfairness in an M/GI/1. In *Proceedings of ACM Sigmetrics 2003 Conference on Measurement and Modeling of Computer Systems*, pages 238-249, San Diego, CA, June 2003.