On the Duality Between Routing and Scheduling Systems With Finite Buffer Space [1]

Panayotis D. Sparaggis, Department of Electrical and Computer Engineering
Christos G. Cassandras, Department of Electrical and Computer Engineering
Don Towsley, Department of Computer and Information Science

University of Massachusetts,
Amherst, MA 01003

ABSTRACT

A duality property is established between scheduling and routing problems associated with a set of parallel queues. This allows one to determine the optimal policy for either system, once it is determined for its dual system. Moreover, the evaluation of different design alternatives (e.g., allocation of buffers) can also be accommodated in the same duality framework. The limitations and the crucial assumptions in establishing the duality properties (e.g., preemptiveness in scheduling policies) are shown. Various applications are presented. It is shown, for instance, that the *Smallest Residual Capacity* scheduling policy is optimal, as being the dual of the well-known *Shortest Queue* routing policy.

March 1992
Submitted to the *IEEE Transactions on Automatic Control*

[1]This work was partly supported by an IBM Graduate Fellowship Award and by NSF under contract ECS-8801912 and ECS-92XXXX.

1 Introduction

Consider the following generic routing and scheduling systems.

- K queues of finite capacity, each with its own server, are fed by a single arrival stream. Customers arrive to a controller which is responsible for routing them to the queues following a predetermined policy.
- K queues of finite capacity, each with its own arrival stream, are served by a single server according to a predetermined policy.

A number of papers have recently appeared that focus on one or the other of these systems. These papers have been primarily concerned with detemining the optimal policy for some variation of one or the other of these systems using either techniques from Markov decision processes [7, 17] or stochastic majorization [14, 15, 19, 20]. It is interesting to observe that the structure of the optimal policies are very similar for these two systems. This suggests that the two problems may be *duals* of each other and that it may be possible to apply results from one problem to the other.

The concept of duality has been frequently introduced and used in the study of various engineering and mathematical problems; a classical application is to convex optimization problems with linear constraints (e.g., [16]). The main idea is to relate a *primal* problem with a properly defined *dual* problem so that any solution to the first immediately generates a solution to the second with no extra computational effort and vice-versa. In some cases, the dual problem is simpler computationally and is directly attacked. Queueing theorists have studied duality properties only in the context of *uncontrolled* systems (e.g., $GI/G/1$ queues [2, 6, 10, 11, 12] and finite-buffer systems [1, 4]). The importance of establishing such properties is twofold. First, it provides a unified view of the nature and the underlying structure of two different problems and enhances our understanding of the system's dynamics. Second, it broadens the set of results known to hold only for one type of systems, by exploiting existing knowledge on their dual counterparts.

In this paper we establish a duality property that couples the state and the performance descriptors between the scheduling and routing systems defined above. Our arguments involve the interchange of the roles of the interarrival and service times and the coupling of the queue lengths in the primal system with the residual queueing capacities in the dual system. The result carries the following important implication. Once a policy has been shown to be optimal in a routing system, a dual (in a sense to be specified in section 2) policy can be immediately claimed to be optimal in a dual scheduling system and vice-versa. The majority of results on controlled systems with *finite* buffer space are very recent and have only been established in a single context (i.e., for routing or scheduling problems). Using the main duality theorem developed in section 2, we are able to show that 'dual' results hold for their counterparts. Moreover, under the assumption that the determined optimal policies are employed, questions regarding parametric perturbations, (e.g. allocation of buffers to queues), are directly answered for both systems using the same duality framework.

Two types of systems are studied, namely, systems in which no buffer space is available at the controller as well as systems in which there is storage capacity at the controller. For the latter type, two particular classes of policies are considered: Earlier Transfer (EX) and Delayed Transfer (DX) policies. The limitations and the crucial assumptions in establishing the duality properties are also demonstrated. It is shown, for instance, that in order to couple the state evolution and the performance measures of routing and scheduling systems with no buffer at the controller and show that one is the dual of the other, one must assume that the scheduling policy is preemptive. On the other hand, when there is buffer space dedicated to the controller, the crucial requirement is that both routing and scheduling policies should be non-idling.

The paper is organized as follows. In section 2 we prove the duality property in systems with no buffer space at the controller and in section 3 we extend the results to systems in which storage space is available to the controller. Then, in section 4 we present a variety of applications on the duality results. It is shown, for instance, that the *Smallest Residual Capacity* scheduling policy is optimal, as being the dual of the well-known *Shortest Queue* routing policy. A discussion of our results is contained in section 5.

2 Duality in systems with no buffer at the controller

In this section we establish a duality property between the two Markovian systems shown in Figure 1. We denote the routing system by $\mathcal{S}^r$ and the scheduling system by $\mathcal{S}^s$. Both systems consist of K queues with finite capacity, labelled $k = 1, 2, ..., K$, but there is no buffer space available to the controller. We assume that queue k in S^a, $a \in \{r, s\}$, can store at most B_k^a jobs, and that B_k^r does include the job (if any) that is being served by the server dedicated to queue k.

In $\mathcal{S}^r$ there exists a server associated with each queue, working under a *non-preemptive* and work-conserving discipline. The service times form K mutually independent sequences of exponential r.v.'s with parameters $\{\mu_k^r\}_{1 \leq k \leq K}$, and are independent of the system's state and the arrival process. There is a single Poisson arrival stream of intensity λ^r and a controller which routes the incoming jobs to the different queues. We consider the class of routing policies Σ^r that have instantaneous queue length information available to them and that are required to route jobs to some queue that has available space, if one exists. Incoming jobs are rejected and lost only if *all* queues are full.

In $\mathcal{S}^s$ the queues are fed by K mutually independent Poisson arrival streams with intensities $\{\lambda_k^s\}_{1 \leq k \leq K}$. All arrival processes are independent of the system's state. There is a single *preemptive* and work-conserving exponential server with parameter μ^s and a controller that schedules jobs from different queues. Similarly to $\mathcal{S}^r$ we focus our attention on the class of scheduling policies Σ^s that have instantaneous state information and that are required to schedule a job from some queue that is not empty, if one exists. If a job that arrives at queue k finds no available space it is rejected and lost.

Let $\mathbf{N}^{\pi_r}(t) = (N_1^{\pi_r}(t), ..., N_K^{\pi_r}(t))$ denote the vector of the queue lengths in $\mathcal{S}^r$ at time $t \geq 0$

under a policy $\pi_r \in \Sigma^r$, i.e., $N_i^{\pi_r}(t)$ denotes the queue length of queue i at time t under π_r, including the customer that is being served. Let $A^{\pi_r}(t)$ denote the number of jobs that are admitted (i.e., not rejected) in $\mathcal{S}^r$ under policy π_r by time t. Furthermore, let $\mathbf{R}^{\pi_s}(t) = (R_1^{\pi_s}(t), ..., R_K^{\pi_s}(t))$ denote the joint residual queueing capacities in $\mathcal{S}^s$ at time $t \geq 0$ under $\pi_s \in \Sigma^s$. That is, $R_i^{\pi_s}(t) = B_i^s - N_i^{\pi_s}(t)$, $i = 1, ..., K$, where $N_i^{\pi_s}(t)$ denotes the queue length of queue i at time t under π_s, including the customer being served (whenever the server visits queue i). Finally, let $C^{\pi_s}(t)$ be the number of jobs that have completed service in $\mathcal{S}^s$ under policy π^s by time t.

We now couple the two systems in the following way:

- $B_k^s = B_k^r, \;\; k = 1, .., K.$
- $\lambda_k^s = \mu_k^r, \;\; , k = 1, .., K.$
- $\mu^s = \lambda^r.$

Our main result now follows. Recall that two real-valued random variables X and Y are said to be equal in a stochastic sense, written $X =_{st} Y$, iff $Pr(X \leq x) = Pr(Y \leq x)$, $x \in I\!R$ (e.g., Ross [13], chapter 8).

Theorem 1 *For any non-idling routing policy $\pi_r \in \Sigma^r$ there exists a preemptive scheduling policy $\pi_s \in \Sigma^s$ such that*

$$\mathbf{N}^{\pi_r}(t) =_{st} \mathbf{R}^{\pi_s}(t) \tag{1}$$
$$A^{\pi_r}(t) =_{st} C^{\pi_s}(t) \tag{2}$$

provided that $\mathbf{N}^{\pi_r}(0) =_{st} \mathbf{R}^{\pi_s}(0)$. Likewise, for any $\pi_s \in \Sigma^s$ there exists a policy $\pi_r \in \Sigma^r$ such that (1), (2) hold, under the same initial condition.

Proof: We prove the first part of the theorem, i.e., the existence of π_s. By symmetry, the second part, i.e., the existence of π_r follows immediately. We condition on arrival times, service times and initial queue lengths. The proof is given by induction on event times $t_0 = 0, t_1, t_2, ...$ We show that

$$\mathbf{N}^{\pi_r}(t) = \mathbf{R}^{\pi_s}(t) \tag{3}$$
$$A^{\pi_r}(t) = C^{\pi_s}(t) \tag{4}$$

on any sample path. We begin with $A^{\pi_r}(0) = C^{\pi_s}(0) = 0$. Furthermore, we assume that $\mathbf{N}^{\pi_r}(0) = \mathbf{R}^{\pi_s}(0)$. Although capital letters are usually reserved to denote random variables, within the proof of this theorem they indicate the values of the variables at specific time instants, on a *single sample path.*

To carry the induction we couple the systems in the following way. First, we couple the service completion times at the kth queue in $\mathcal{S}^r$ with the arrival times at the kth queue in $\mathcal{S}^s$

under the two policies. This is possible since $\lambda_k^s = \mu_k^r$ and the distributions are exponential. Furthermore, if a queue is empty in $\mathcal{S}^r$ we assume that the server serves a fictitious customer and that a customer that arrives to that queue receives the remainder of this service time. The exponential assumption here is required to guarantee that all true times form a sequence of i.i.d. exponential r.v.'s. We also couple the arrival times in $\mathcal{S}^r$ with the departure times in $\mathcal{S}^s$. This is possible since $\lambda^r = \mu^s$ and the distributions are exponential. If all queues in $\mathcal{S}^s$ are empty the server serves fictitious customers, similar to $\mathcal{S}^r$. Finally, π_s is defined in the following way. At any time t, π_s schedules the server to the same queue as the one to which π_r would route an incoming customer if an arrival occurred at that time t in $\mathcal{S}^r$. Since service times in $\mathcal{S}^s$ are exponential we may assume that when the server in $\mathcal{S}^s$ preempts a job and switches to another queue, the newly scheduled job receives the remainder of the service time of the job that was originally being served. These preemption instants (under π_s) are exactly those at which π_r changes the index of the queue to which the next arriving job is to be routed.

Basic step. By the statement of the Theorem, relations (3), (4) hold for $t = t_0$.

Inductive step. Assume that the two relations hold up through $t = t_n$. Clearly they hold for $t_n < t < t_{n+1}$. For t_{n+1} we consider the following cases.

Case 1. Service completion in $\mathcal{S}^r$.
Suppose that the next event is a completion at the kth queue in $\mathcal{S}^r$ at time t_{n+1}. This implies that an arrival occurs at queue k in $\mathcal{S}^s$ at t_{n+1}. Our induction hypothesis $N_k^{\pi_r}(t_n) = R_k^{\pi_s}(t_n)$ implies that the queue length in queue k of $\mathcal{S}^s$ will be increased by one iff the service completion is a real one in $\mathcal{S}^r$. In other words, if the service completion is fictitious in $\mathcal{S}^r$ then $N_k^{\pi_r}(t_n) = 0$, thus $R_k^{\pi_s}(t_n) = 0$, so the customer who arrives at the kth queue in $\mathcal{S}^s$ is rejected. Therefore,

$$N_k^{\pi_r}(t_{n+1}) = N_k^{\pi_r}(t_n) - \mathbf{1}(N_k^{\pi_r}(t_n) > 0) = R_k^{\pi_s}(t_n) - \mathbf{1}(R_k^{\pi_s}(t_n) > 0) = R_k^{\pi_s}(t_{n+1}).$$

This yields (3) at t_{n+1}. As to equation (4) it follows trivially at t_{n+1} since,

$$A^{\pi_r}(t_{n+1}) = A^{\pi_r}(t_n) = C^{\pi_s}(t_n) = C^{\pi_s}(t_{n+1}).$$

Case 2. Arrival in $\mathcal{S}^r$.
Suppose that the next event on the sample path is an arrival of a customer in $\mathcal{S}^r$. This implies that a service completion takes place in $\mathcal{S}^s$ at t_{n+1}. Due to the coupling of the two policies, if a customer is routed to queue k in $\mathcal{S}^r$ under π_r, then a customer completes service at queue k in $\mathcal{S}^s$ under π_s. Thus, (3) follows trivially at t_{n+1}. Note that the coupling of the two policies becomes possible since by the induction hypothesis, and the fact that $B_k^s = B_k^r$, we have,

$$N_k^{\pi_r}(t_n) = R_k^{\pi_s}(t_n) \Rightarrow \mathbf{1}(N_k^{\pi_r}(t_n) < B_k^r) = \mathbf{1}(R_k^{\pi_s}(t_n) < B_k^s) \tag{5}$$

As to equation (4) we have,

$$A^{\pi_r}(t_{n+1}) = A^{\pi_r}(t_n) + \mathbf{1}(\sum_{i=1}^{K} N_i^{\pi_r}(t_n) < \sum_{i=1}^{K} B_i^r) = C^{\pi_s}(t_n) + \mathbf{1}(\sum_{i=1}^{K} R_i^{\pi_s}(t_n) < \sum_{i=1}^{K} B_i^s) = C^{\pi_s}(t_{n+1}) \tag{6}$$

The middle equality in the above relation follows from the induction hypothesis $\mathbf{N}^{\pi_r}(t_n) = \mathbf{N}^{\pi_s}(t_n)$ which implies $\sum_{i=1}^{K} N_i^{\pi_r}(t_n) = \sum_{i=1}^{K} R_i^{\pi_s}(t_n)$, and the fact $B_k^s = B_k^r$ for all $k \in \{1, .., K\}$. This completes the inductive step.

Removal of the conditioning on arrival times, service times and initial queue lengths completes the theorem. ∎

Remark 1.
It is important to note that the server in $\mathcal{S}^s$ may preempt a job and switch to another queue at any time. Naturally, allowing for preemption in scheduling systems leads to optimal policies that outperform non-preemptive, *idling* policies (since the latter class of policies is a contained in the former), which in turn perform better than non-preemptive, work-conserving policies. On the other hand, it is also evident that there is no reason to preempt jobs in $G/M/1/K$ systems, if one is interested in maximizing throughput (or equivalently, minimize losses) because of the memoryless property of the exponential distribution[2]. Since each individual queue in $\mathcal{S}^r$ can be treated as a $G/M/1/K$ queue (where the arrival statistics depend on the routing policy), it is clear that the assumption of non-preemptiveness does not hurt the system's throughput. As it turns out, in the proof of Theorem 1 it has actually been *necessary* to assume that the queueing discipline in $\mathcal{S}^s$ is preemptive. Specifically, if the queueing discipline in $\mathcal{S}^s$ were non-preemptive then it would become impossible to show that equation (3) propagates along the sample path. In particular, in case 2 of the proof it would not be possible to guarantee that a job completes service in $\mathcal{S}^s$ at the same queue to which a job is routed in $\mathcal{S}^r$. The requirement for preemption in the scheduling system agrees with intuition. Indeed, if the server in $\mathcal{S}^s$ were assumed to be non-preemptive, then we would attempt to couple scheduling systems that employ *suboptimal* policies (as compared to preemptive policies which are optimal) with routing systems that employ optimal control procedures (preemption here is not critical), which is counterintuitive.

Remark 2.
Interchanging the role of interarrival and service times is the central theme in duality theory. The specific arguments that have been used fall into two categories. The first approach involves comparisons of state and other random variables (such as hitting times) using algebraic manipulations (e.g., [10, 11, 12]). The second methodology is based on sample path comparisons (e.g., [2] and Takacs, [18] chapter 5) as in Theorem 1 above. The latter approach provides valuable insight into the dynamics of dual systems and is more intuitive. Time reversibility (see Kelly [8], a standard reference on the topic) is sometimes a necessary tool in the construction involved in the sample path methodology. See for example [2], where it was shown that one may equate the time reversed path of the residency time (of the customer in the server) in a $GI/N/1$ queue with a path of the workload in the dual $N/GI/1$ queue, where N denotes the Neuts process. Time reversibility was not needed in the proof of Theorem 1 above, because of the assumption of exponential distributions (see also section 6 of [2] for another example).

Remark 3.

[2]If however, one is interested in other performance metrics, in particular delays, then preemptive policies (e.g., LIFO) may be optimal.

One of the main themes in Theorem 1 is that of coupling the queue lengths in the primal system with the residual capacities in the dual system. The idea has sometimes been referred to as the 'customer-hole' duality (e.g., [1, 4, 5]), where a hole means an empty space in the buffer. This can be thought of as a *dynamic-type* duality, since it prescribes a duality on the dynamics of the state trajectories. A *static-type* duality can be defined to pertain only to the input parameters of dual systems. This second type of duality was present in Theorem 1 (as well as in [1, 4]), where the role of the mean service and the mean interarrival times had to be interchanged in moving from the primal routing system to its dual scheduling system.

We now move to the study of steady-state performance measures. Consider the classes of stationary routing and scheduling policies which induce ergodic Markov chains in the corresponding systems. Thus, it makes sense to define the asymptotic average performance measures,

$$\phi^{\pi_a} = \lim_{t\to\infty}(A^{\pi_a}(t)/t), \quad ,\pi_a \in \Sigma^a, a \in \{r,s\} \tag{7}$$

$$\theta^{\pi_a} = \lim_{t\to\infty}(C^{\pi_a}(t)/t), \quad ,\pi_a \in \Sigma^a, a \in \{r,s\} \tag{8}$$

$$\tag{9}$$

which are independent of the initial state. Note that we have extended our notation to include $A^{\pi_s}(t)$, i.e., the number of customers that enter $\mathcal{S}^s$ by time t under π_s, and $C^{\pi_r}(t)$, i.e., the number of customers that complete service in $\mathcal{S}^r$ by time t under π_r. Since both systems have finite capacities, it further follows that $\phi^{\pi_a} = \theta^{\pi_a}$, $a \in \{r,s\}$. Thus a policy π_a^* that maximizes ϕ^{π_a} over the class of admissible policies also maximizes θ^{π_a} and vice-versa.

A stationary policy $\pi_r \in \Sigma^r$ is defined as $\pi_r = f(N_1^r, ..., N_K^r)$, with $f : I\!N^K \to \{1, .., K\}$, $I\!N = \{0, 1, 2, ..., \}$. We define a *dual* policy $\pi_s \in \Sigma^s$, and denote $\pi_s = (\pi_r)^D$, to be induced by the mapping $\pi_s = f(R_1^r, ..., R_K^r)$ for the same function f. The following corollary can be established easily.

Corollary 1 *Let π_r^* be a policy in Σ^r that maximizes ϕ^{π_r} (θ^{π_r}) over all policies $\pi_r \in \Sigma^r$. It follows that the dual policy of π_r^* maximizes ϕ^{π_s} (repectively θ^{π_s}) over all policies π_s in Σ^s, i.e.,*

$$\pi_s^* = (\pi_r^*)^D.$$

Proof: It follows from Theorem 1 that the policy that maximizes ϕ^{π_r} in $\mathcal{S}^r$ gives rise to a dual policy that maximizes θ^{π_s} in $\mathcal{S}^s$. Since $\theta^{\pi_s} = \phi^{\pi_s}$ this latter policy also maximizes ϕ^{π_s} over all policies in Σ^s, and the rest of the proof is straightforward. ∎

Remark 4.
Another interesting class of systems is that in which no state information is available to the controller. For instance, routing can be random or round-robin, and the assumption is that queue lengths cannot be observed. Clearly, the same duality property holds for this class of systems, since the proof of Theorem 1 makes no assumptions about how π_s routes customers to queues. Hence, one may well assume that π_s makes no use of state information. Note, however,

that since an arrival to a full queue in the routing system is coupled with scheduling the server to an empty queue in the dual scheduling system, the server in the latter system may indeed stay idle at a queue even at times when there exist other non-empty queues. In other words, it should be assumed that when a queue is visited by the server in the scheduling system, a whole service period is taken (corresponding to the service time needed by a single customer) regardless of whether that particular queue is empty or non-empty. One may think, for instance, of a polling system in which the time to switch the server from one station to another dominates the customer's own service time. Other examples can be thought in the context of time/frequency reservation schemes (e.g., slotted time protocols) employed in communication networks.

3 Duality in systems with buffer space at the controller

This section studies systems in which some finite buffer space is available to the controller. To extend the notation of the previous section, in addition to the K parallel queues labelled $k = 1, ..., K$ there exists another queue dedicated to the controller labelled $k = 0$ with finite capacity $B_0^r = B_0^s$ (see figure 2). We assume that B_0^s includes the server. Similarly, we extend the state definition so that $\mathbf{N}^{\pi_r}(t)$ and $\mathbf{R}^{\pi_s}(t)$ become $(K+1)$-dimensional vectors, i.e., $\mathbf{N}^{\pi_r}(t) = (N_0^{\pi_r}(t), N_1^{\pi_r}(t), ..., N_K^{\pi_r}(t))$ and $\mathbf{R}^{\pi_s}(t) = (R_0^{\pi_s}(t), R_1^{\pi_s}(t), ..., R_K^{\pi_s}(t))$.

We are interested in studing two classes of *non-idling* control policies, namely, the class of Earlier Transfer (EX) policies which we denote by Σ_{EX} and the class of Delayed Transfer (DX) policies which we denote by Σ_{DX}. This terminology was first introduced in [20]. Note, however, that the class of DX policies studied here (for which duality holds), is somewhat different than the class of DX policies studied in [20]. We will postpone the discussion of the latter class to the end of this section. As previously, we use the superscripts r, s to distinguish routing from scheduling systems, e.g., we denote the class of EX policies admissible in $\mathcal{S}^r$ by Σ_{EX}^r.

Let us now see how these policies are defined. A policy in Σ_{EX}^r transfers a job from queue 0 to one of queues $1, ..., K$ as soon as it arrives unless all queues $1, ..., K$ are full, in which case it retains the job at queue 0 until a space becomes available. Similarly, a policy in Σ_{EX}^s transfers a job to queue 0 as soon as it arrives to one of queues $1, ..., K$, provided that there is some available space at queue 0; otherwise, it retains the job at the queue where it arrived, until such space becomes available.

On the other hand, a policy in Σ_{DX}^r transfers a job from queue 0 to one of queues $1, ..., K$, say queue k, only if queue k is empty. Otherwise, it retains all arriving jobs at queue 0 until one of queues $1, ..., K$ becomes empty unless queue 0 itself becomes full, in which case it immediately transfers one job to one of queues $1, ..., K$ (determined by the policy π_r) that has available space (if such a queue exists) in order to prevent future overflow at queue 0 (by emptying one buffer space at queue 0). Similarly, a policy in Σ_{DX}^s transfers a job to queue 0 only when this queue becomes empty unless one of queues $1, ..., K$ becomes full, say queue k, in which case it transfers one job from queue k to queue 0 (provided that queue 0 is not full) to prevent future overflow at queue k.

Having defined the classes of Earliest and Delayed Transfer policies as above one can easily extend Theorem 1.

Theorem 2 *For any non-idling routing policy $\pi_r \in \Sigma^r_{EX}$ (Σ^r_{DX}) there exists a non-idling scheduling policy $\pi_s \in \Sigma^s_{EX}$ (resp. Σ^s_{DX}) such that*

$$\mathbf{N}^{\pi_r}(t) =_{st} \mathbf{R}^{\pi_s}(t) \tag{10}$$

$$A^{\pi_r}(t) =_{st} C^{\pi_s}(t) \tag{11}$$

provided that $\mathbf{N}^{\pi_r}(0) =_{st} \mathbf{R}^{\pi_s}(0)$. Likewise, for any $\pi_s \in \Sigma^s_{EX}$ (Σ^s_{DX}) there exists a policy $\pi_r \in \Sigma^r_{EX}$ (resp. Σ^r_{DX}) such that (10), (11) hold, under the same initial condition.

Proof. The proof of this theorem is similar to that of Theorem 1. The only additional feature of the proof lies in ensuring that $N_0^{\pi_r}(t) = R_0^{\pi_s}(t)$ for all t on a single path. This can be easily shown to be true given how EX and DX are defined. ∎

Implicit in the definition of Σ_{DX} is the assumption that a job is lost when it arrives in front of a full queue and cannot directly be transferred to either one of the queues $1, ..., K$ in $\mathcal{S}^r$, or, to queue 0 in $\mathcal{S}^s$, if some available space exists in those queues. Although this is a very natural assumption it is of interest to see what happens if a job can be transferred directly upon arrival. In this case, it is possible to have queue 0 in $\mathcal{S}^r$ being full while at least one of the queues $1, ..., K$ is non-full. Hence, it is also possible to have queue 0 in $\mathcal{S}^s$ being empty while at least one of queues $1, ..., K$ is non-empty, thus, keeping the server idle. In fact, this has been the setting of DX scheduling policies in [20]. It is interesting to note that in this case it is not possible to couple the state descriptors $\mathbf{N}^{\pi_r}(t)$ and $\mathbf{R}^{\pi_s}(t)$ over all sample paths. To see this, first notice that $\mathbf{N}^{\pi_r}(t) = \mathbf{R}^{\pi_s}(t)$ implies $\sum_{i=0}^K N_i^{\pi_r}(t) = \sum_{i=0}^K R_i^{\pi_s}(t)$ at any time instant t. Now, if for example $N_0^{\pi_r}(t_n) = B_0^r = B_0^s = R_0^{\pi_s}(t_n)$, $N_1^{\pi_r}(t_n) = R_1^{\pi_s}(t_n) < B_1^s$,..., $N_K^{\pi_r}(t_n) = R_K^{\pi_s}(t_n) < B_K^s$, and the next event is an arrival at $\mathcal{S}^r$ (thus a service completion at $\mathcal{S}^s$) occurring at time t_{n+1}, it is seen that the job will be accepted in $\mathcal{S}^r$ (by bypassing queue 0), whereas no job will leave $\mathcal{S}^s$ at t_{n+1} (i.e., the service completion in $\mathcal{S}^s$ is fictitious). Thus,

$$\sum_{i=0}^K N_i^{\pi_r}(t_{n+1}) = \sum_{i=0}^K N_i^{\pi_r}(t_n) + 1 \neq \sum_{i=0}^K N_i^{\pi_r}(t_n) = \sum_{i=0}^K R_i^{\pi_s}(t_n) = \sum_{i=0}^K R_i^{\pi_s}(t_{n+1}). \tag{12}$$

Hence, $\mathbf{N}^{\pi_r}(t_{n+1}) \neq \mathbf{R}^{\pi_s}(t_{n+1})$. This shows that Theorem 2 holds only for non-idling policies.

4 Applications

This section describes some applications of the duality property described in Theorem 1. Our analysis gives rise to dual results for problems that have been studied either in a routing or scheduling setting only. It is also interesting to note that duality properties can be exploited in analyzing problems that have not been thoroughly studied yet.

The first application considers the duality of the optimal policies between routing systems with *equal-rate* servers and scheduling systems with *equal-rate* arrival streams. Recently, the intuitive *Join the Shortest Non-Full Queue* (SNQ) policy was shown ([19]) to maximize (in a strong stochastic sense) the number of customers which are lost due to buffer overflow, in a routing system that consists of queues with unequal finite capacities, servers of equal rate and no buffer at the controller. Under the SNQ policy, a job is routed to the queue with the least number of customers that is not full. Moreover, although queues with equal queue lengths may have unequal residual capacities and therefore are dinstict, it was shown that ties can be broken arbitrarily.

Since the SNQ policy is optimal regardless of the initial state, Theorem 1 implies that the policy that always schedules a job from the queue with the smallest residual capacity maximizes the number of customers that enter service by any time t, in the dual scheduling system that consists of queues with equal-rate arrival streams. Clearly, this policy is the dual of the SNQ policy. We call it the *Smallest Residual Capacity* (SRC) policy. Ties are again broken arbitrarily although queues with the same residual capacities may have unequal queue lengths.

On the other hand, when there is buffer space available at the controller, the SRC policy was shown to be optimal [20] over the class Σ_{EX}^{s} for scheduling systems with equal capacities at queues $1, ..., K$, in the context of flow control in gateways between low speed LANs and a high speed MAN. Thus, the SNQ policy is optimal in Σ_{EX}^{r}. In fact, following the arguments in [15], one expects that the SRC and SNQ policies remain optimal when capacities at queues $1, ..., K$ are unequal. When the service rates are queue-dependent the structure of the optimal scheduling policies in a two-queue system has been studied in [17] (the optimal policies were shown to be of switching type). The cost structure in [17] is very general and accounts for both holding and blocking costs. Theorem 1 can be applied in the special case in which holding costs are zero and blocking costs for the two queues are the same. In this case, the optimal policy for the dual routing system is of switching type.

A related problem is to determine the optimal allocation of B buffers to queues $1, ..., K$ in a routing or scheduling system. In the case of a scheduling system, it should also be assumed that exactly one buffer has been already assigned to each queue, so as to make sure that exactly, and not less than, K queues will be finally formed after the allocation of the 'extra' B buffers. Let B_i denote the buffers assigned to queue i, $i = 1, \cdots, K$, under some feasible allocation scheme $\mathbf{B} = (B_1, \cdots, B_i, \cdots, B_K)$. Define an allocation scheme $\mathbf{B}^b = (B_1^b, \cdots, B_K^b)$ (where the superscript stands for 'balanced') such that

$$B_i^b = \begin{cases} \lfloor B/K \rfloor + 1, & B \bmod K \neq 0, i = 1, .., B \bmod K, \\ \lfloor B/K \rfloor & \text{otherwise.} \end{cases} \tag{13}$$

i.e., B_i^b's can differ by one at most. In [19], the above scheme was shown to be optimal in the sense of minimizing the number of rejected customers (equivalently, maximizing the number of admitted customers) up to any time t when no buffer is available at the controller, provided that all considered feasible schemes employ the optimal SNQ policy. Thus, it is implied by Theorem 1 that B^b outperforms all alternative allocation schemes in scheduling systems, provided again that in all cases the optimal SRC policy (which is the dual of the SNQ policy) is followed.

A straightforward generalization of Theorem 1 is to consider routing systems with state dependent service rates $\mu_k^r(\mathbf{N})$, as well as scheduling systems with state-dependent arrival rates $\lambda_k^s(\mathbf{R})$. In the context of routing systems with equal-rate exponential servers and no buffer at the controller two results have been reported in [14]. When service rates are concave and non-decreasing in the queue lengths and buffer capacities are unequal then the SNQ policy stochastically minimizes the number of rejected customers by any time t. On the other hand, when service times are convex and non-decreasing in the queue lengths, then the *Longest Non-Full Queue* (LNQ) policy is optimal with respect to the same performance metric, provided that queue capacities are equal. As understood, the LNQ policy routes a customer to the queue with the longest queue length that is not at capacity. These results give rise to optimal policies in scheduling systems with arrival rates that are non-decreasing convex or concave in the residual capacities. Furthermore, the duality property applies to the buffer allocation problem as shown in Table 1. In this table, $\mathbf{B}^{ub}$ denotes the *unbalanced* allocation where all buffers are assigned to a single queue, and LRC stands for the *Longest Residual Capacity* policy. Some problems that have not been studied so far, admit solutions that are natural extensions of already reported results, and they can be studied within a duality framework. For instance, one expects that the SNQ and SRC policies are also optimal over the classes Σ_{DX}^r and Σ_{DX}^s respectively.

As mentioned in the previous section, the results of Theorem 1 can be directly applied to the study of systems in which no state information is available to the controller. *Routing* systems of this type were studied in [15], where it was shown that the *Round-Robin* policy stochastically minimizes the number of losses by any time t under the assumptions of exponential equal-rate servers and of queue capacities being equal. Hence, the Round-Robin policy is also optimal in the dual scheduling system for the same performance criterion. In routing systems with Bernoulli splitting it can be shown that assigning equal probabilities of splitting the arrivals into the various queues minimizes losses, when all servers are identical (the complete proof of this result will be reported elsewhere). This generalizes similar results on queues with infinite capacities (e.g., [3]). Again, it is clear that the same Bernoulli assignment is optimal when employed in scheduling systems with probabilistic control schemes.

Last, consider the problem of controlling the arrival and service rates in routing systems and scheduling systems respectively. For example, assume that the optimal allocation of a fixed total arrival rate $\lambda = \lambda_1^s + \cdots + \lambda_K^s$ among the various queues of a scheduling system is to be determined. An example is distributing inventory over a number of physically distinct store locations (or warehouses) which are served by a single transportation vehicle, used to transfer inventory to various selling stations. The problem is complex in its general setting, where queues have unequal capacities and the optimal policy for each allocation may not be known. Clearly, the dual problem is how to distribute a total processing power $\mu = \mu_1^r + \cdots + \mu_K^r$ among the queues of a routing system. Theorem 1 implies that the optimal splitting of the traffic is identical to the optimal splitting of the processing power. Moreover, comparisons (possibly through analytical tools such as queueing theory) among different allocation schemes for a scheduling system translates into comparisons among different distribution patterns for a routing system. This facilitates determining a 'good' (if not optimal) distribution. Note that in scheduling systems the problem can also be thought of as the one of distributing a number of S identical servers among the K queues, with $S \geq K$, whereas in routing systems one may

think of the problem of aggregating S separate arrival streams into K sessions.

5 Discussion

The idea of coupling queue lengths with residual capacities can also be exploited within a more theoretical framework, in particular in studying properties of majorization as they arise in the modeling of routing and scheduling systems. Consider two K-dimensional vectors $\mathbf{M}, \mathbf{N}$. The definition of majorization $\mathbf{M} \prec \mathbf{N}$ is motivated as a way of making precise the idea that the components of $\mathbf{M}$ are 'less spread-out' than the components of $\mathbf{N}$. Thus, majorization can be used to compare different degrees of load-balancing, a feature inherent in optimization problems arising in routing and scheduling systems.

Consider, for example, the two weak forms of majorization, namely, *weak submajorization* (denoted '$\prec_w$') and *weak supermajorization* (denoted $\prec^w$'). To keep the discussion brief, we refer to [9] for definitions. Let $\hat{M}_k, \hat{N}_k$ be the kth largest element in $\mathbf{M}, \mathbf{N}$ respectively. It was shown in [15] that if $\mathbf{M} \prec_w \mathbf{N}$ then it is true, $(\hat{M}_1, ..., max(\hat{M}_m - 1, 0), ..., \hat{M}_K) \prec_w (\hat{N}_1, ..., max(\hat{N}_n - 1, 0), ..., \hat{N}_K$, for $m \leq n$. The same is true for '$\prec^w$'. Think of the vectors $\mathbf{M}, \mathbf{N}$ as representing queue lengths. Then, the two properties before imply the preservation of weak sub/super majorization under departure operators. Now assume that both $\mathbf{M}, \mathbf{N}$ are bounded from above by B, the queue capacity in a scheduling or a routing system in which all of the queues have the same available buffer space. The key observation is that similar preservation properties can be shown to hold under operators that pertain to arrival events, via the use of the residual capacities. For example, it is true that $\mathbf{M} \prec_w \mathbf{N}$ implies $(\hat{M}_1, ..., min(\hat{M}_m + 1, B), ..., \hat{M}_K) \prec_w (\hat{N}_1, ..., min(\hat{N}_n + 1, B), ..., \hat{N}_K$, for $m \geq n$. To show this, define $M_i^R = B - M_i$ and $N_i^R = B - N_i$, $i = 1, ..., K$ and observe that,

$$\begin{aligned} & \mathbf{M} \prec_w \mathbf{N} \Rightarrow \mathbf{M}^R \prec^w \mathbf{N}^R \Rightarrow \\ \Rightarrow \quad & (\hat{M}_1^R, ..., max(\hat{M}_m^R - 1, B), ..., \hat{M}_K^R) \prec^w (\hat{N}_1^R, ..., max(\hat{N}_n^R - 1, 0), ..., \hat{N}_K^R), m \leq n \Rightarrow \\ \Rightarrow \quad & (\hat{M}_1, ..., min(\hat{M}_m + 1, B), ..., \hat{M}_K) \prec_w (\hat{N}_1, ..., min(\hat{N}_n + 1, B), ..., \hat{N}_K), m \geq n. \end{aligned} \quad (14)$$

The use of arrival operators was hidden under various algebraic arguments in the proofs of [14] as well as in [19, 20], although the associated preservation properties follow in a straighforward way, as shown above. Other less intuitive properties of majorization can also be studied in the same duality framework.

References

[1] Ammar M.H and S.B.Gershwin (1989). Equivalence relations in queueing models of Fork/Join networks with blocking. *Performance Evaluation* 10: 233-245.

[2] Asmussen S. and V.Ramaswami (1990). Probabilistic interpretations of some duality results for the matrix paradigms in queueing theory. *Stochastic Models* 6: 715-713.

[3] Chang C.S. (1990). A new ordering for stochastic majorization: theory and applications. *IBM Research Report 16028.*

[4] Dallery Y., Z. Liu and D. Towsley (1991). Equivalence, reversibility and concavity properties in Fork/Join queueing networks with blocking. *COINS Technical Report 91-88.* University of Massachusetts.

[5] Gordon W. J. and G.F.Newell (1967). Cyclic queueing networks with restricted length queues. *Operations Research* 15: 266-278.

[6] Heathcote C.R. (1965). On the maximum of the queue $GI/M/1$. *Journal of Applied Probability* 2: 206-214.

[7] Hordijk A. and G.Koole (1990). On the optimality of the generalized shortest queue policy. *Probability in the Engineering and Information Sciences* 4: 477-487.

[8] Kelly F.P. (1979). *Reversibility and stochastic networks.* New York: Wiley.

[9] Marshall A.W. and I.Olkin (1979). *Inequalities: Theory of majorization and its applications.* Academic Press.

[10] Ramaswami V. (1990). A duality theorem for the matrix paradigms in queueing theory. *Stochastic Models* 6: 161-161.

[11] Ramaswami V. and M.F.Neuts (1980). A duality theorem for phase-type queues. *Annals of Probability* 8: 974-985.

[12] Roes P.B.M. (1973). Two finite queues and their duals. *Journal of Applied Probability* 5: 22-24.

[13] Ross S.M. (1983). *Stochastic processes.* New York: Wiley.

[14] Sparaggis P.D., D.Towsley and C.G.Cassandras (1991). Extremal properties of the Shortest Non-Full Queue policy and the Longest Non-Full Queue policy in finite capacity systems with state-dependent service rates. To appear in the *Journal of Applied Probability.*

[15] Sparaggis P.D., D.Towsley and C.G.Cassandras (1991). Optimality of static routing policies in queueing systems with blocking. Submitted for publication.

[16] Stoer J. and C. Witzgall (1970). *Convexity and optimization in finite dimensions.* New York: Springer-Verlag.

[17] Suk J.B. and C.G.Cassandras (1991). Optimal scheduling of two competing queues with blocking. *IEEE Transactions on Automatic Control* 36: 1086-1091.

[18] Takacs L. (1967). *Combinatorial methods in the theory of stochastic processes.* New York: Wiley.

[19] Towsley D., P.D.Sparaggis and C.G.Cassandras (1991). Optimal routing and buffer allocation for a class of finite capacity queueing systems. To appear in the *IEEE Transactions on Automatic Control.*

[20] Towsley D., S.Fdida, H.Santoso (1991). Design and evaluation of flow control protocols for Metropolitan Area Networks. In G.Pujjole (ed.), *Architecture and performance issues of high-capacity local and metropolitan area networks*: pp. 471-492. New York: Springer Verlag.

Table 1. Some applications of the duality property (DP).

	Routing systems	*Scheduling systems*
No buffer at the controller, unequal capacities, constant arrival/service rate	SNQ is optimal, Th. 1 [15]	SRC is optimal, DP
Buffer at the controller, equal capacities, constant arrival/service rate	SNQ is optimal over Σ^r_{EX}, DP	SRC is optimal over Σ^r_{EX}, Th. 3 [20]
Buffer allocation, no buffer at the controller, constant arrival/service rate	$\mathbf{B}^o$ is optimal Th. 2 [15]	$\mathbf{B}^o$ is optimal, DP
No buffer at the controller, unequal capacities, concave arrival/service rate	SNQ is optimal, Th. 1 [14]	SRC is optimal, DP
Buffer allocation, no buffer at the controller concave arrival/service rate	$\mathbf{B}^b$ is optimal, Th. 5 [14]	$\mathbf{B}^b$ is optimal, DP
No buffer at the controller, equal capacities, convex arrival/service rate	LNQ is optimal, Th. 3 [14]	LRC is optimal, DP
Buffer allocation, no buffer at the controller convex arrival/service rate	$\mathbf{B}^{ub}$ is optimal, Th. 6 [14]	$\mathbf{B}^{ub}$ is optimal, DP

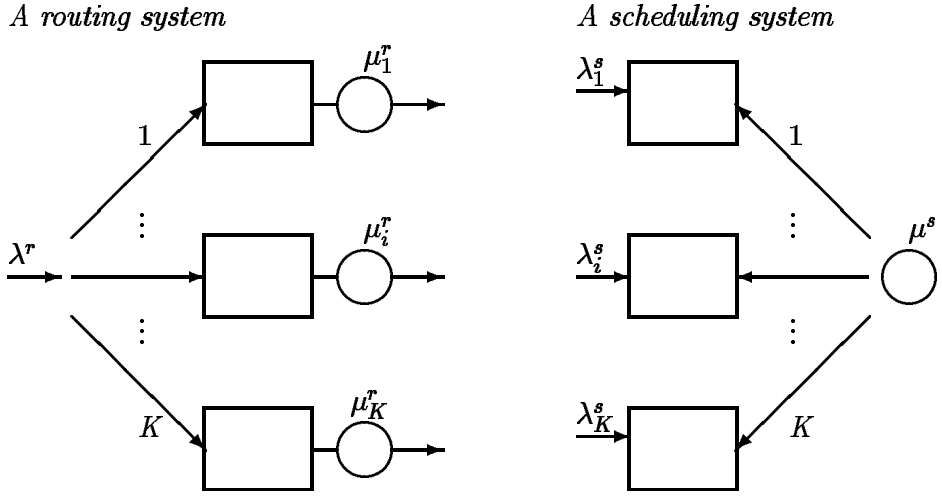

Figure 1: A routing (S^r) and a scheduling (S^s) system with K queues

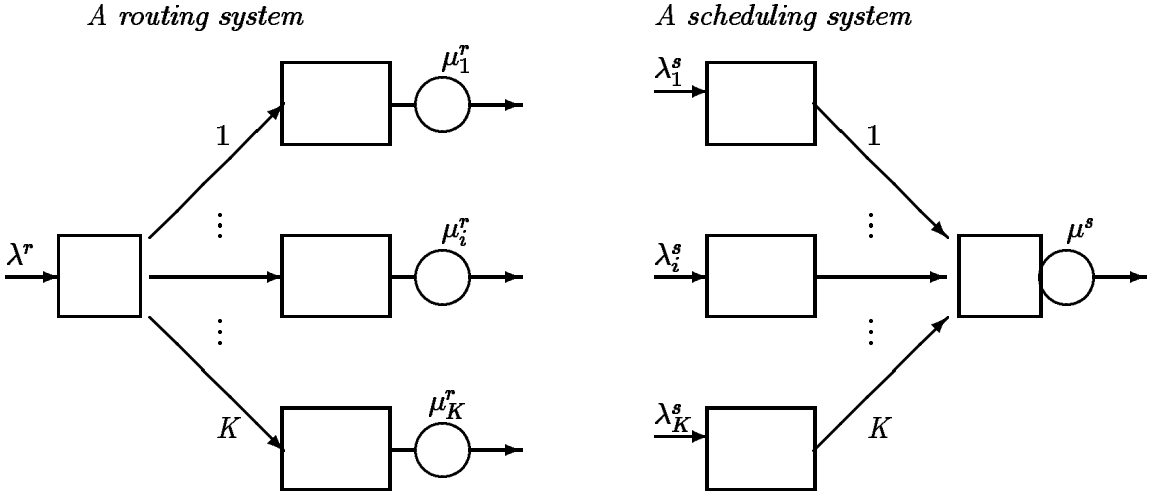

Figure 2: A routing (S^r) and a scheduling (S^s) system with K queues and buffer space at the controller